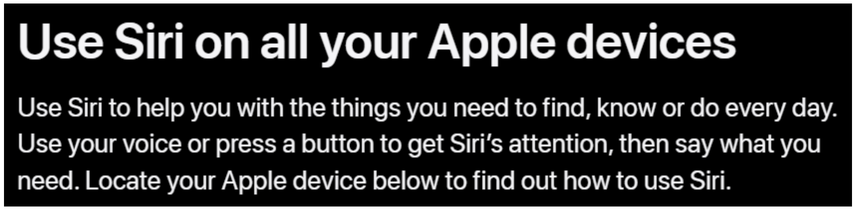
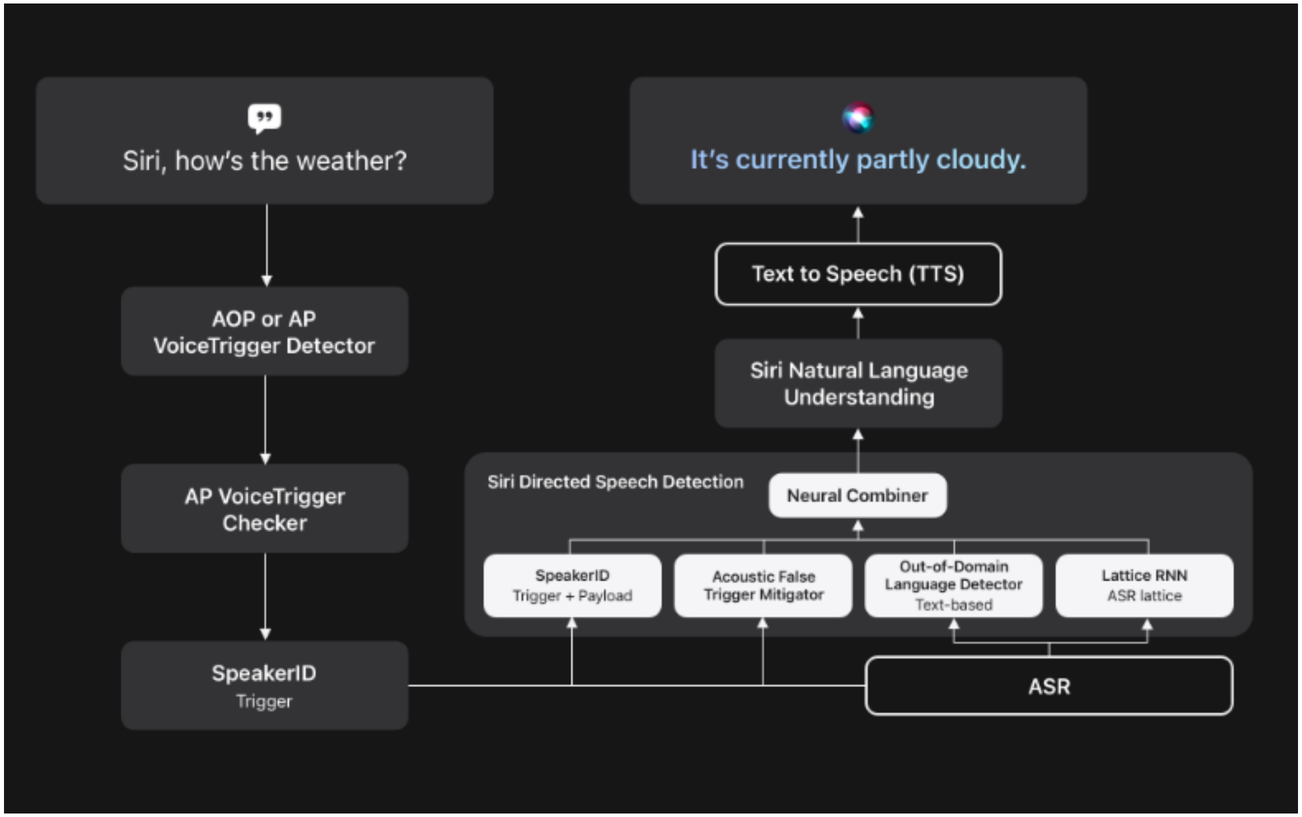
EXHIBIT B

U.S. Patent No. US 8,521,766 v. Apple Inc.

1. Claim Chart

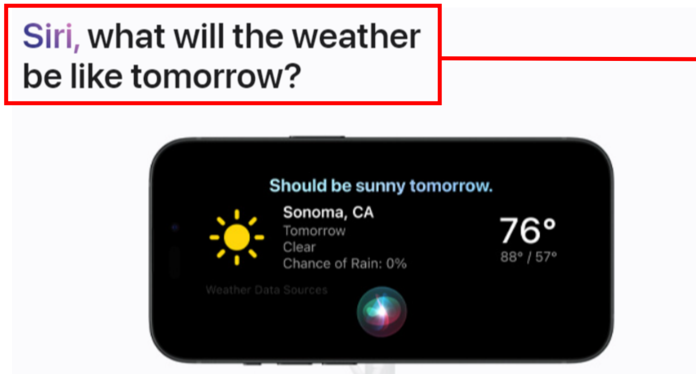
Claim	Analysis
[1.P] A method, comprising	Apple ("Company") performs and/or induces others to perform a method. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Company provides Siri, an intelligent voice assistant that receives voice commands from a user through a mobile device such as an iPhone and retrieves information related to the voice command.  Use Siri on all your Apple devices Use Siri to help you with the things you need to find, know or do every day. Use your voice or press a button to get Siri's attention, then say what you need. Locate your Apple device below to find out how to use Siri. Source: https://support.apple.com/en-us/105020

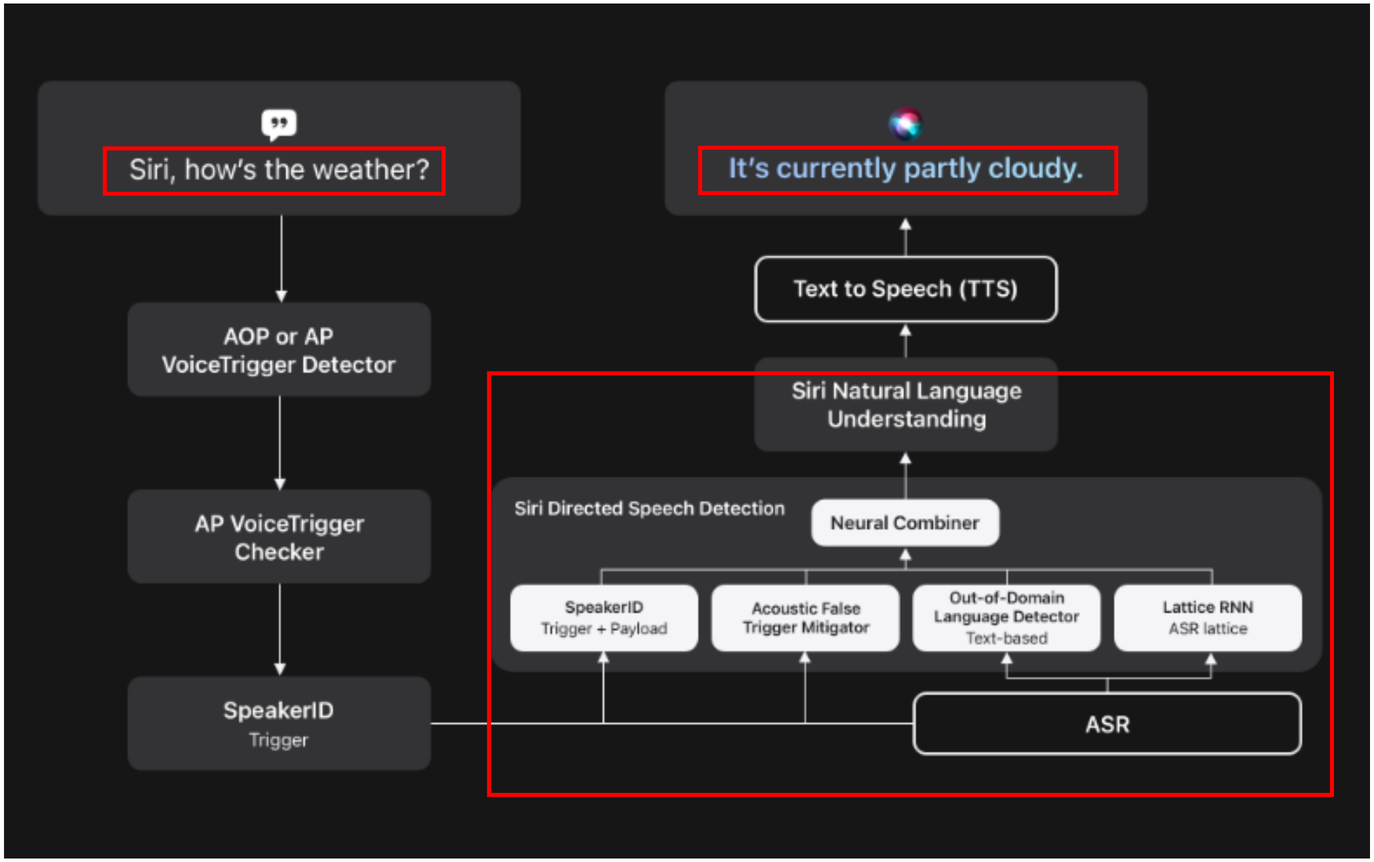
<table>
<tr><td></td><td>

Source: https://machinelearning.apple.com/research/voice-trigger#figure1

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.</td></tr>
<tr><td>[1.1] receiving an information request</td><td>Company performs and/or induces others to perform a step of receiving an information request.

This element is infringed literally, or in the alternative, under the doctrine of equivalents.</td></tr>
</table>

	For example, Siri receives an input voice command (“information request”) given by the user through their mobile devices. The command comprises a trigger phrase and a subsequent utterance, the trigger phrase being ‘Siri’ or ‘Hey Siri’.  information request Source: https://www.apple.com/siri/ (annotated) In this article, we will discuss how Apple has designed a high-accuracy, privacy-centric, power-efficient, on-device voice trigger system with multiple stages to enable natural voice-driven interactions with Apple devices. The voice trigger system supports several Apple device categories like iPhone, iPad, HomePod, AirPods, Mac, Apple Watch, and Apple Vision Pro. Apple devices simultaneously support two keywords for voice trigger detection: “Hey Siri” and “Siri.” Source: https://machinelearning.apple.com/research/voice-trigger Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[1.2] decoding the	Company performs and/or induces others to perform a step of decoding the information request.

information request;	This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri uses automatic speech recognition and natural language processing to process the information request received from the user. It checks whether the user command is directed towards Siri or not and then, identifies the intent of the command using a Siri Directed Speech Detection (SDSD) system. The SDSD comprises various False Trigger Mitigation (FTM) systems such as Acoustic FTM, Out-of-domain Language Detector, and Lattice RNN which decode the input command and convert it into the intent. 

	Source: https://machinelearning.apple.com/research/voice-trigger#figure1 Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled). Source: https://machinelearning.apple.com/research/hey-siri When a voice trigger detection mechanism detects a trigger, the system starts processing user audio using a full-blown ASR system. A dedicated algorithm determines the end-of-speech event, at which point we obtain the ASR output and the decoding lattice. We use word-aligned lattices such that each arc corresponds to Source: https://machinelearning.apple.com/research/voice-trigger model scores, text, etc. NLU signals are comprised of domain classification features such as domain categories, domain scores, sequence labels of the user request transcription, etc. An intent is a combination of ASR and NLU signals. We refer to these signals as *understanding signals* decoded by ASR and NLU sub-systems. Every intent is encoded into a vector space and this process is described in Section 4.1. Our task is to produce a ranked list of intents using information-state in addition to understanding signals to choose the best response. Source: https://arxiv.org/pdf/2005.00119.pdf, Page 2

	Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[1.3] discovering information using the decoded information request;	Company performs and/or induces others to perform a step of discovering information using the decoded information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, after the intents ("decoded information request") are determined, Siri searches ("discovering information") for the relevant information.

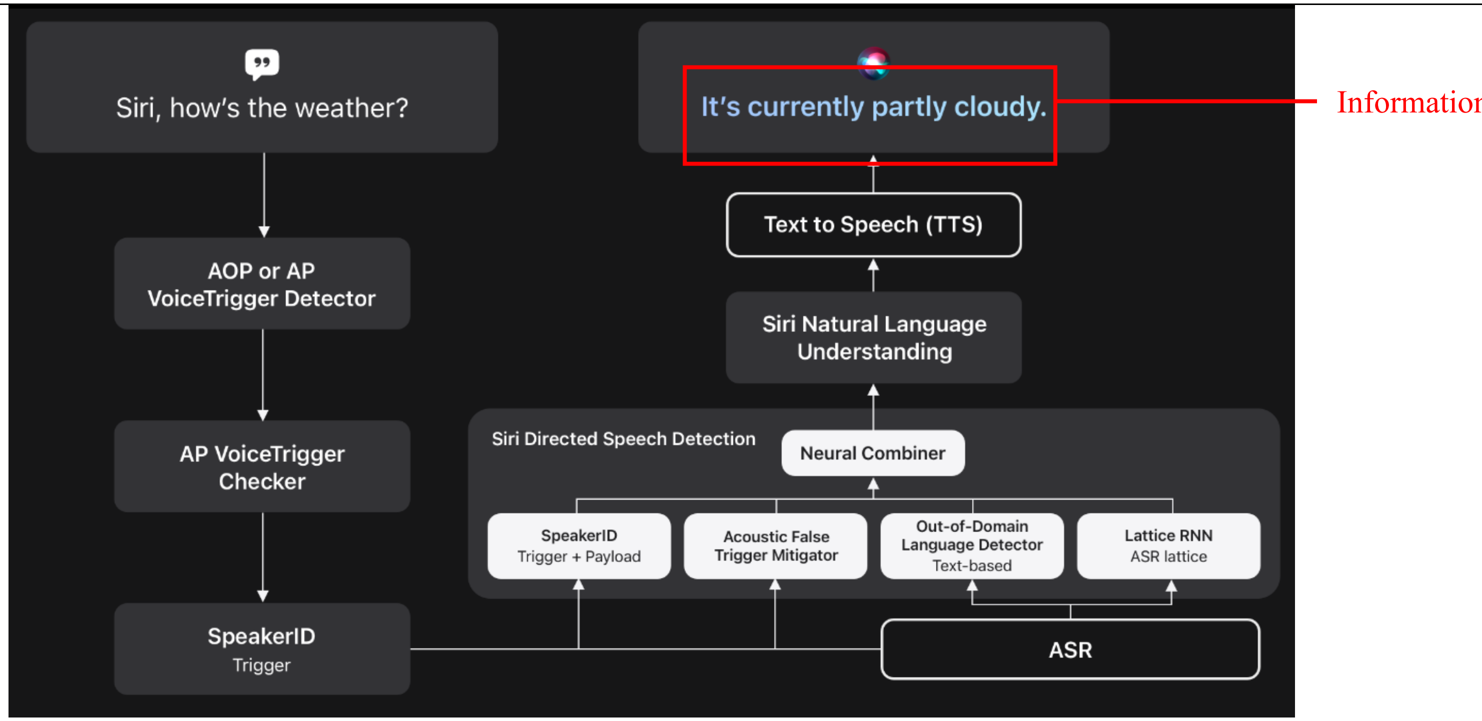
SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

discovering information

Source: https://developer.apple.com/design/human-interface-guidelines/siri (annotated)

	Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled). Source: https://machinelearning.apple.com/research/hey-siri

<table>
<tr>
<td></td>
<td>


Source: https://machinelearning.apple.coms/research/voice-trigger (annotated)

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
</td>
</tr>
<tr>
<td>[1.4] preparing, using one or more processing devices instructions for accessing</td>
<td>
Company performs and/or induces others to perform a step of preparing, using one or more processing devices instructions for accessing the information.

This element is infringed literally, or in the alternative, under the doctrine of equivalents.

For example, after converting audio requests to the intents, Siri provides conversational flow which helps the apps to fulfill the user request according to the domain of the intents and displays the relevant information. Therefore, it would
</td>
</tr>
</table>

the information, the instructions including:	be apparent to a person having ordinary skill in the art that Siri prepares instructions for accessing the information using one or more processing devices. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis. Source: https://developer.apple.com/design/human-interface-guidelines/siri/ **System intents** SiriKit defines a large number of system intents that represent common tasks people do, such as playing music, sending messages to friends, and managing notes. For system intents, Siri defines the conversational flow, while your app provides the data to complete the interaction. Source: https://developer.apple.com/design/human-interface-guidelines/siri#System-intents

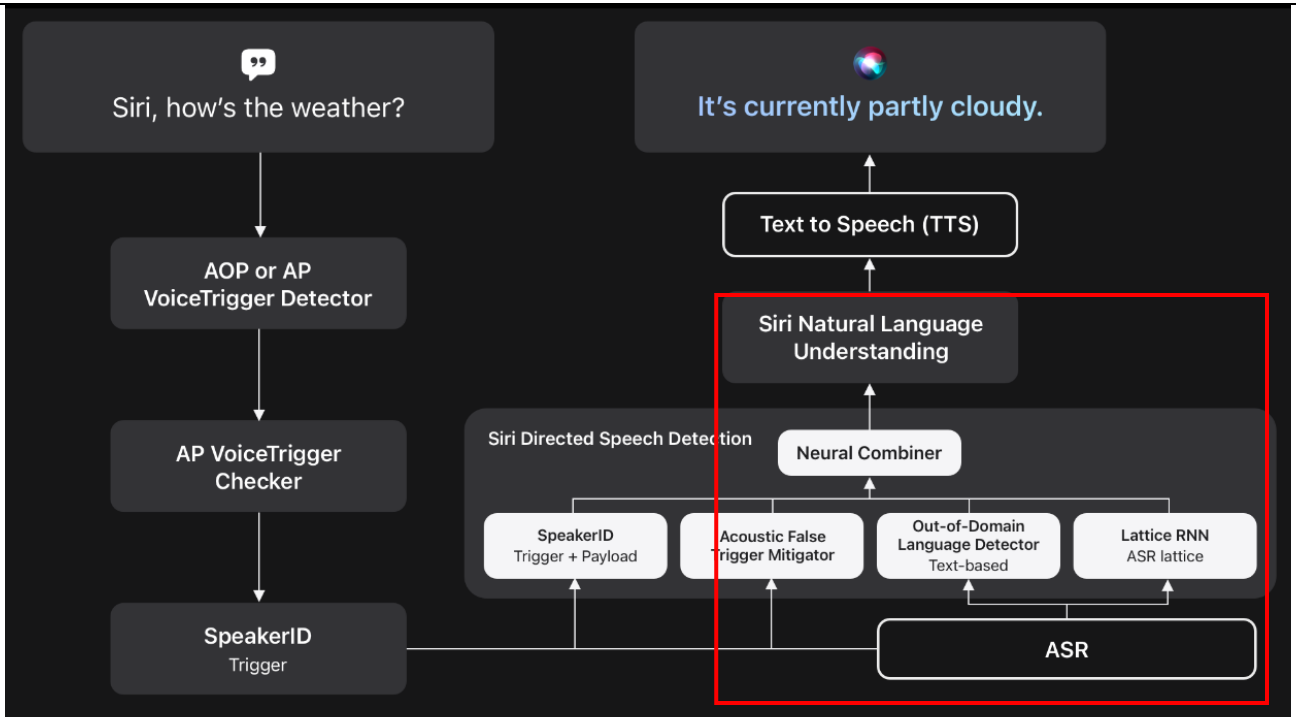
SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

Source: https://developer.apple.com/design/human-interface-guidelines/siri

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[1.5] one or more Automatic Speech Recognition (ASR) grammar codes;	Company performs and/or induces others to perform a step of preparing, using one or more processing devices instructions for accessing the information, the instructions including: one or more Automatic Speech Recognition (ASR) grammar codes. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri does the language processing and semantic analysis to convert the requests into the intents. During semantic analysis, the audio input is matched against a grammar ("one or more Automatic Speech Recognition (ASR) grammar codes") to produce a semantic interpretation of the input. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis. Source: https://developer.apple.com/design/human-interface-guidelines/siri/ **1.4 Semantic Interpretation** A speech recognizer is capable of matching audio input against a grammar to produce a *raw text* transcription (also known as *literal text*) of the detected input. A recognizer may be capable of, but is not required to, perform subsequent processing of the raw text to produce a *semantic interpretation* of the input. Source: https://www.w3.org/TR/2004/REC-speech-grammar-20040316/#S1.3

	 Source: https://machinelearning.apple.coms/research/voice-trigger Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[1.6] one or more short text string matching codes; and	Company performs and/or induces others to perform a step of preparing, using one or more processing devices instructions for accessing the information, the instructions including: one or more short text string matching codes. This element is infringed literally, or in the alternative, under the doctrine of equivalents.

For example, Siri does the natural language processing and semantic analysis to convert the requests into the intents. Further, Siri matches the data with string to retrieve relevant result. Since, the relevant information is retrieved according to the intent, upon information and belief, the instructions comprise one or more short text string matching codes.

A closer look at intents

When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis.

Source: https://developer.apple.com/design/human-interface-guidelines/siri/

Overview

Your app likely defines a number of custom types that model the data the app creates or consumes. For example, a music app might define types that represent artists, albums, and tracks. Because those types are unique to your app, the framework can't interpret them until you expose them to system services such as Siri and the Shortcuts app. *Entities* are lightweight types that provide information to the system about your app's data or concepts relating to that data. An entity identifies and queries the data it represents and describes how the system displays that data onscreen.

Source: https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents

	To let people use arbitrary text to find specific entities, adopt the `EntityStringQuery` protocol instead. Queries that adopt this protocol cause the system to display a search field above the list of suggested entities. Implement the required `entities(matching:)` function, and use the provided string to match against your data. For example, a music app might let people search for a specific album by matching against the album name. Source: https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[1.7] one or more information formatting codes operative to format a consumer device display; and	Company performs and/or induces others to perform a step of preparing, using one or more processing devices instructions for accessing the information, the instructions including: one or more information formatting codes operative to format a consumer device display. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, the intents describe how the system displays ("information formatting codes operative to format a consumer device display") the data such as dates, times, and addresses. **Overview** Your app likely defines a number of custom types that model the data the app creates or consumes. For example, a music app might define types that represent artists, albums, and tracks. Because those types are unique to your app, the framework can't interpret them until you expose them to system services such as Siri and the Shortcuts app. *Entities* are lightweight types that provide information to the system about your app's data or concepts relating to that data. An entity identifies and queries the data it represents and describes how the system displays that data onscreen.

	Source: https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents Siri displays entities like dates, times, addresses and currency amounts in a nicely formatted way. This is the result of the application of a process called inverse text normalization (ITN) to the output of a core speech recognition component. To understand the important role ITN plays, consider that, without it, Siri would display "October twenty third twenty sixteen" instead of "October 23, 2016". In this work, we Source: https://machinelearning.apple.com/research/inverse-text-normal Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[1.8] communicating the prepared instructions.	Company performs and/or induces others to perform a step of communicating the prepared instructions. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri communicates the instructions to execute intents to the apps in the user's mobile device such as iPhone such that the relevant information is accessed. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis. Source: https://developer.apple.com/design/human-interface-guidelines/siri/

SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

Source: https://developer.apple.com/design/human-interface-guidelines/siri

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

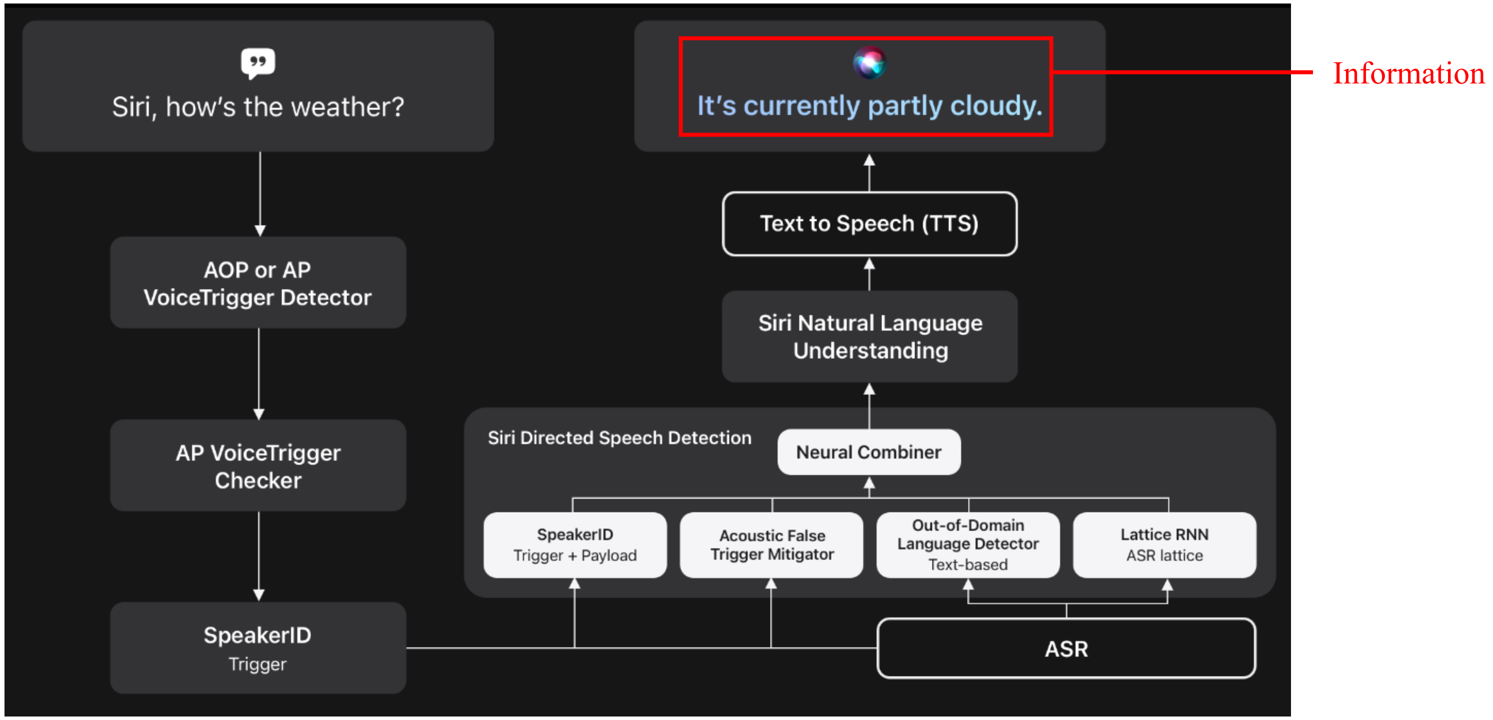
<table>
<tr>
<td>[2] The method of claim 1, wherein the discovering information using the decoded information request comprises:</td>
<td>
Company performs and/or induces others to perform the method of claim 1, wherein the information is discovered using the decoded information request.

This element is infringed literally, or in the alternative, under the doctrine of equivalents.

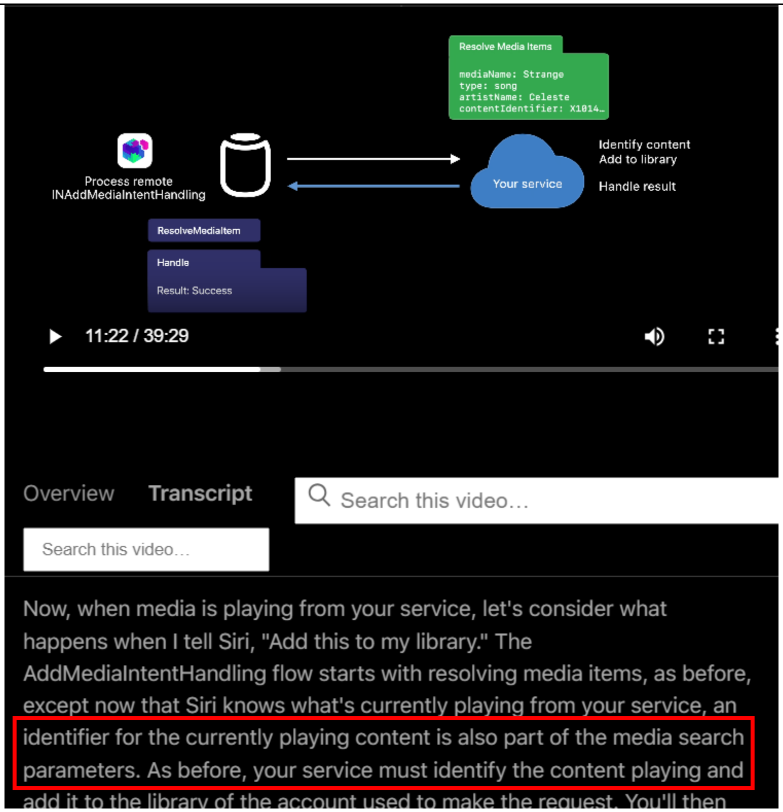
For example, after the intents (“decoded information request”) are determined, the relevant information is retrieved based on the intent.

SiriKit provides the following intents.
<table>
<tr><th>Domain (link to developer guidance)</th><th>Intents</th></tr>
<tr><td>VoIP Calling</td><td>Initiate calls.</td></tr>
<tr><td>Workouts</td><td>Start, pause, resume, end, and cancel workouts.</td></tr>
<tr><td rowspan="3">Lists and Notes</td><td>Create notes.</td></tr>
<tr><td>Search for notes.</td></tr>
<tr><td>Create reminders based on a date, time, or location.</td></tr>
<tr><td rowspan="3">Media</td><td>Search for and play media content, such as video, music, audiobooks, and podcasts.</td></tr>
<tr><td>Like or dislike items.</td></tr>
<tr><td>Add items to a library or playlist.</td></tr>
</table>
information is discovered using the decoded information request
</td>
</tr>
</table>

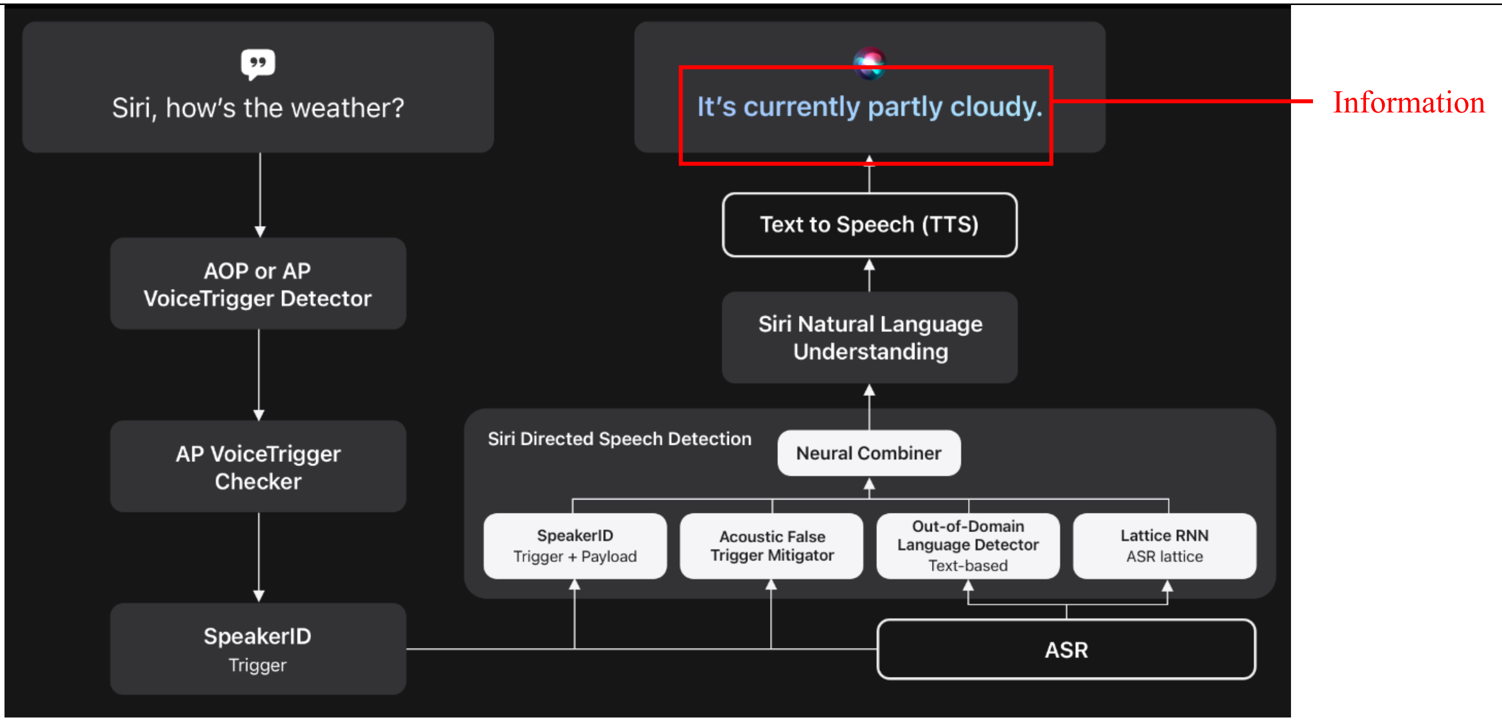
	Source: https://developer.apple.com/design/human-interface-guidelines/siri (annotated) Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled). Source: https://machinelearning.apple.com/research/hey-siri

	 Source: https://machinelearning.apple.coms/research/voice-trigger (annotated) Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[2.1] accessing a database, the database configured at least for containing	Company performs and/or induces others to perform the step of claim 1, wherein the discovering information using the decoded information request comprises: accessing a database, the database configured at least for containing references to information available from a third party source. This element is infringed literally, or in the alternative, under the doctrine of equivalents.

references to information available from a third party source; and	For example, Siri uses intents to retrieve relevant information such that the information is displayed to the user. For instance, if a user asks about the weather, Siri retrieves the weather-related information from the third party weather sources and displays the relevant information. Therefore, upon information and belief, a database is accessed that contains references to information available from a third-party source. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis. Source: https://developer.apple.com/design/human-interface-guidelines/siri/



Source: https://developer.apple.com/videos/play/tech-talks/10854/ at 11:22.



Source: https://machinelearning.apple.coms/research/voice-trigger (annotated)

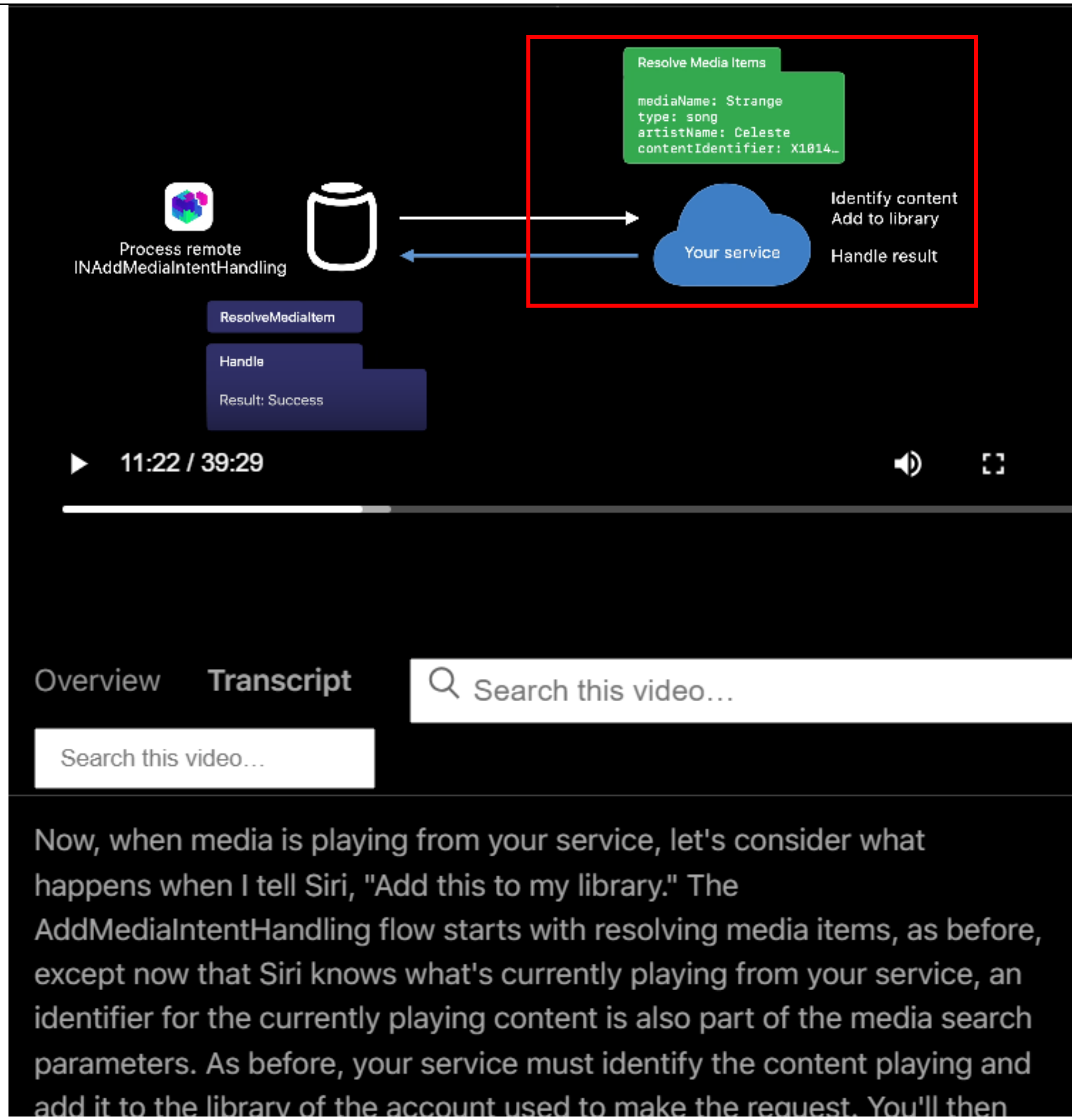
SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

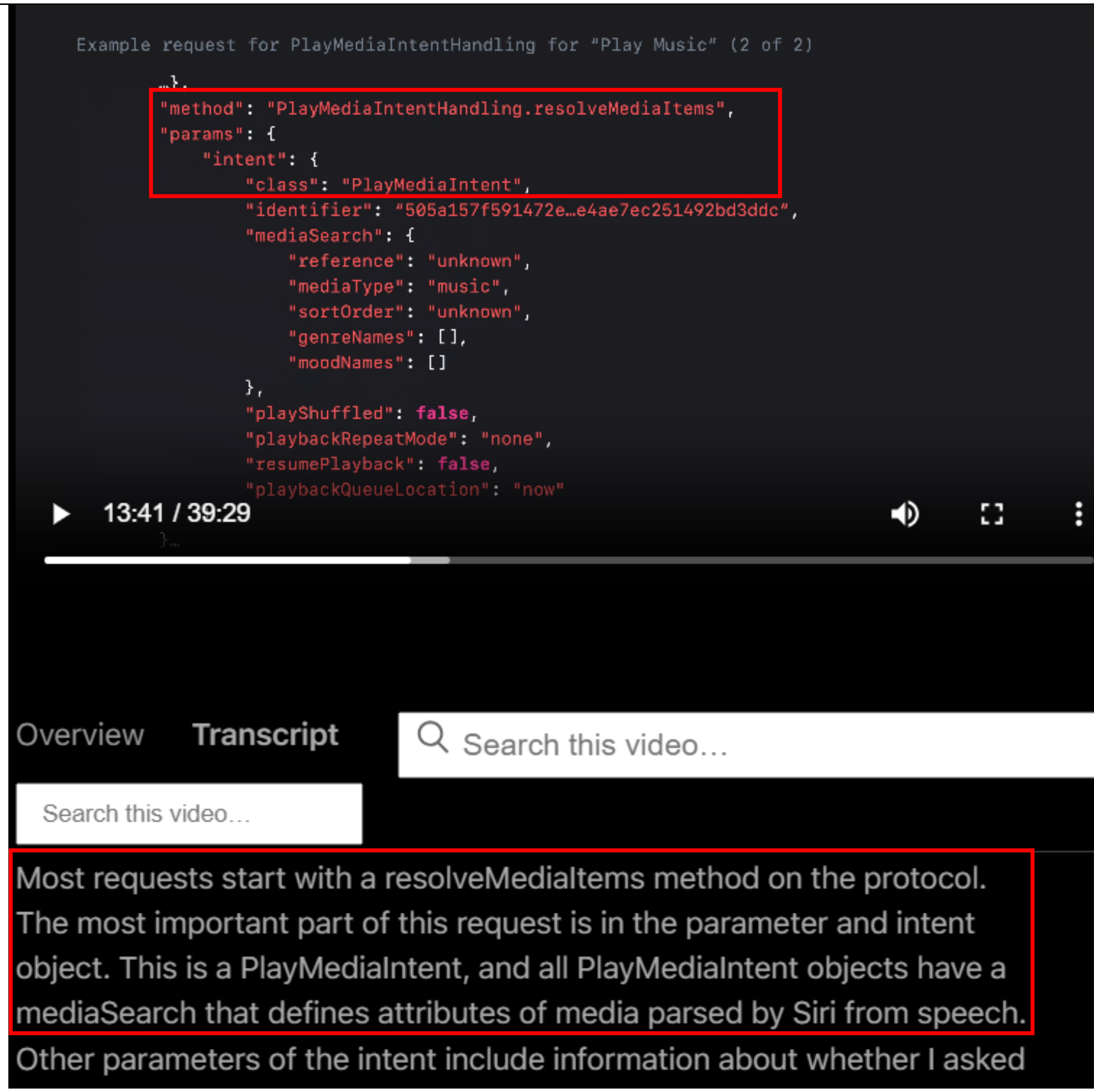
Source: https://developer.apple.com/design/human-interface-guidelines/siri

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[2.2] identifying a reference to the information.	Company performs and/or induces others to perform the step of claim 1, wherein the discovering information using the decoded information request comprises: identifying a reference to the information. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri uses intents to retrieve relevant information such that the information is displayed to the user. Therefore, upon information and belief, a database is accessed that contains references to information available from a third-party source.



Source: https://developer.apple.com/videos/play/tech-talks/10854/ at 11:22.



Source: https://developer.apple.com/videos/play/tech-talks/10854/ at 13:41.

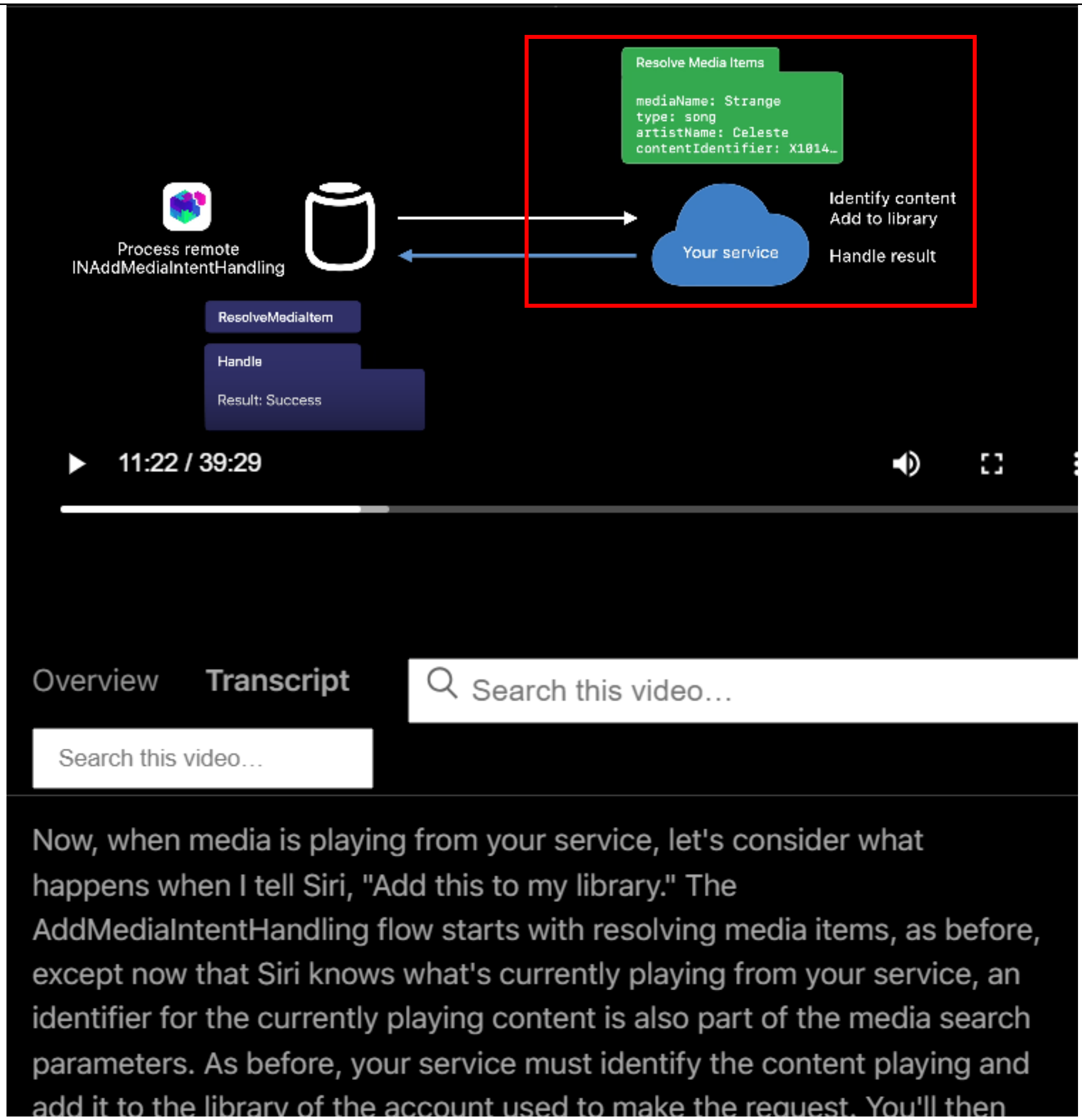
	Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[3] The method of claim 2, wherein the preparing, using one or more processing devices, instructions for accessing the discovered information comprises: preparing instructions for accessing the referenced information from the third party source.	Company performs and/or induces others to perform a method of claim 2, wherein the preparing, using one or more processing devices, instructions for accessing the discovered information comprises: preparing instructions for accessing the referenced information from the third party source. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, after converting audio requests to the intents, Siri provides conversational flow for system intents such that the information is retrieved. These flows help the app to fulfill the user request according to the domain of the intents. Therefore, upon information and belief, Siri prepares a set of instructions for accessing the referenced information from the third-party source. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis. Source: https://developer.apple.com/design/human-interface-guidelines/siri/

	System intents SiriKit defines a large number of system intents that represent common tasks people do, such as playing music, sending messages to friends, and managing notes. For system intents, Siri defines the conversational flow, while your app provides the data to complete the interaction. Source: https://developer.apple.com/design/human-interface-guidelines/siri#System-intents

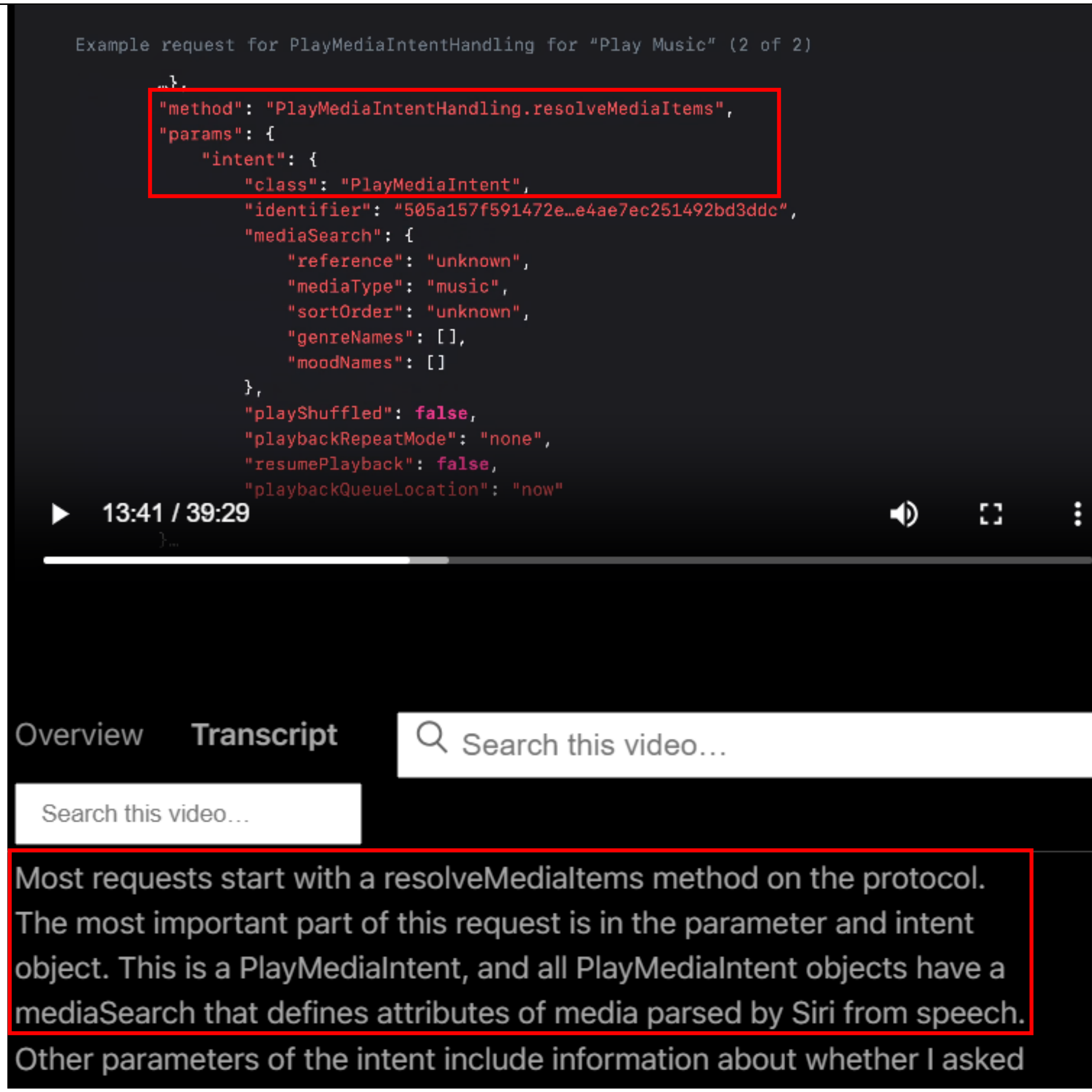
	SiriKit provides the following intents. Domain (link to developer guidance) / Intents VoIP Calling: Initiate calls. Workouts: Start, pause, resume, end, and cancel workouts. Lists and Notes: Create notes. / Search for notes. / Create reminders based on a date, time, or location. Media: Search for and play media content, such as video, music, audiobooks, and podcasts. / Like or dislike items. / Add items to a library or playlist. Source: https://developer.apple.com/design/human-interface-guidelines/siri

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

Source: https://developer.apple.com/design/human-interface-guidelines/siri

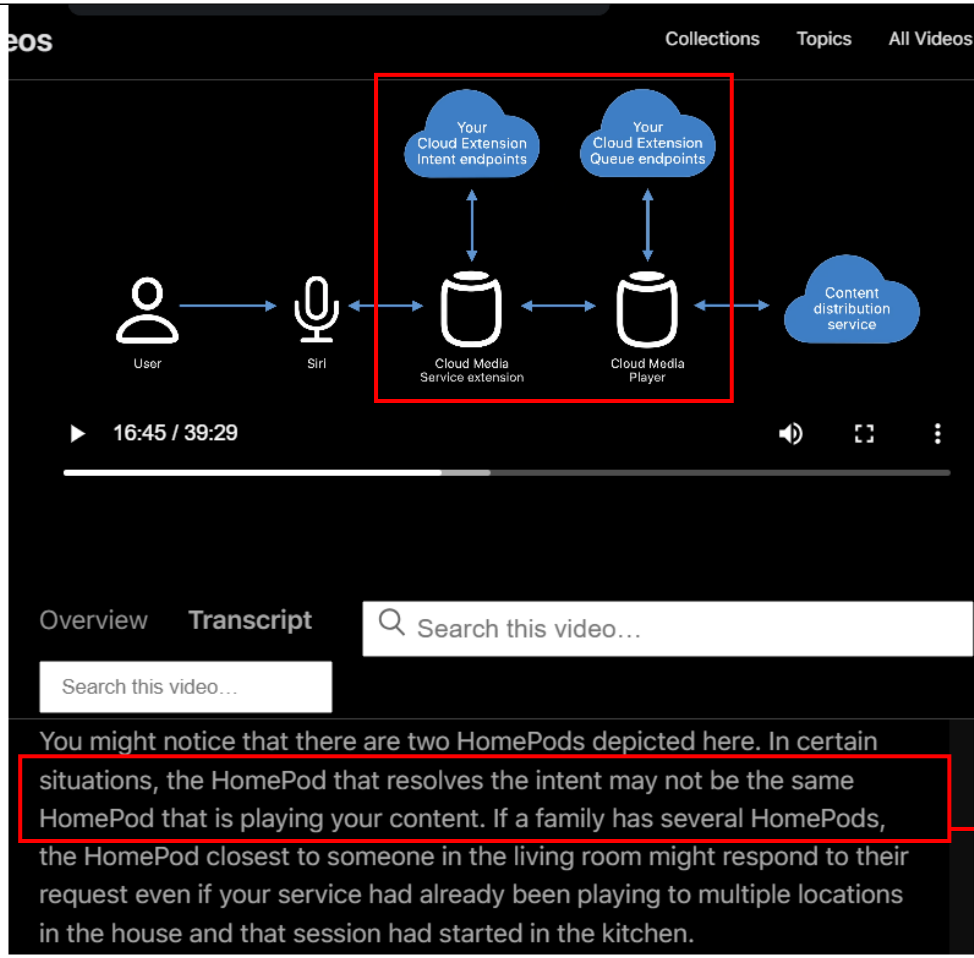


Source: https://developer.apple.com/videos/play/tech-talks/10854/ at 11:22.



Source: https://developer.apple.com/videos/play/tech-talks/10854/ at 13:41.

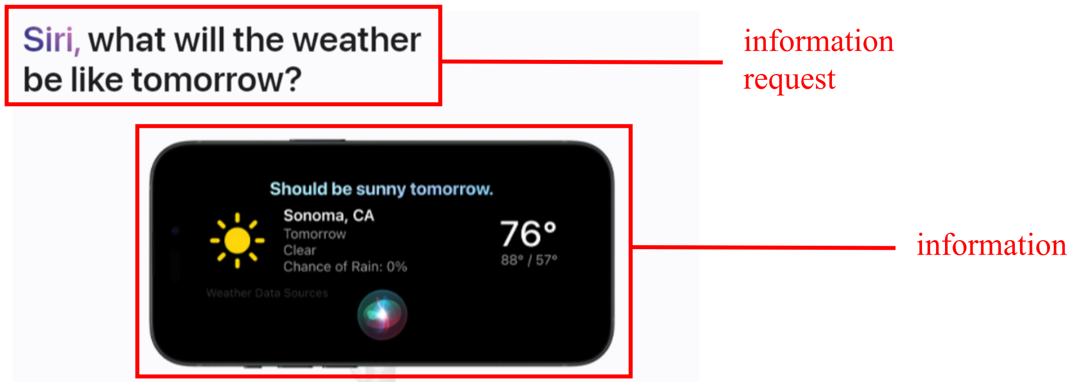
	Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[5] The method of claim 1, further comprising communicating the prepared instructions to a consumer device different from a requesting consumer device.	Company performs and/or induces others to perform the method of claim 1, further comprising communicating the prepared instructions to a consumer device different from a requesting consumer device. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri allows to take voice input intent from one iOS device (“requesting consumer device”), resolve the intent on the same device, and plays the content (“communicating the prepared instructions”) based on the intent on another iOS device ("consumer device different from a requesting consumer device”), that is different from the first device.

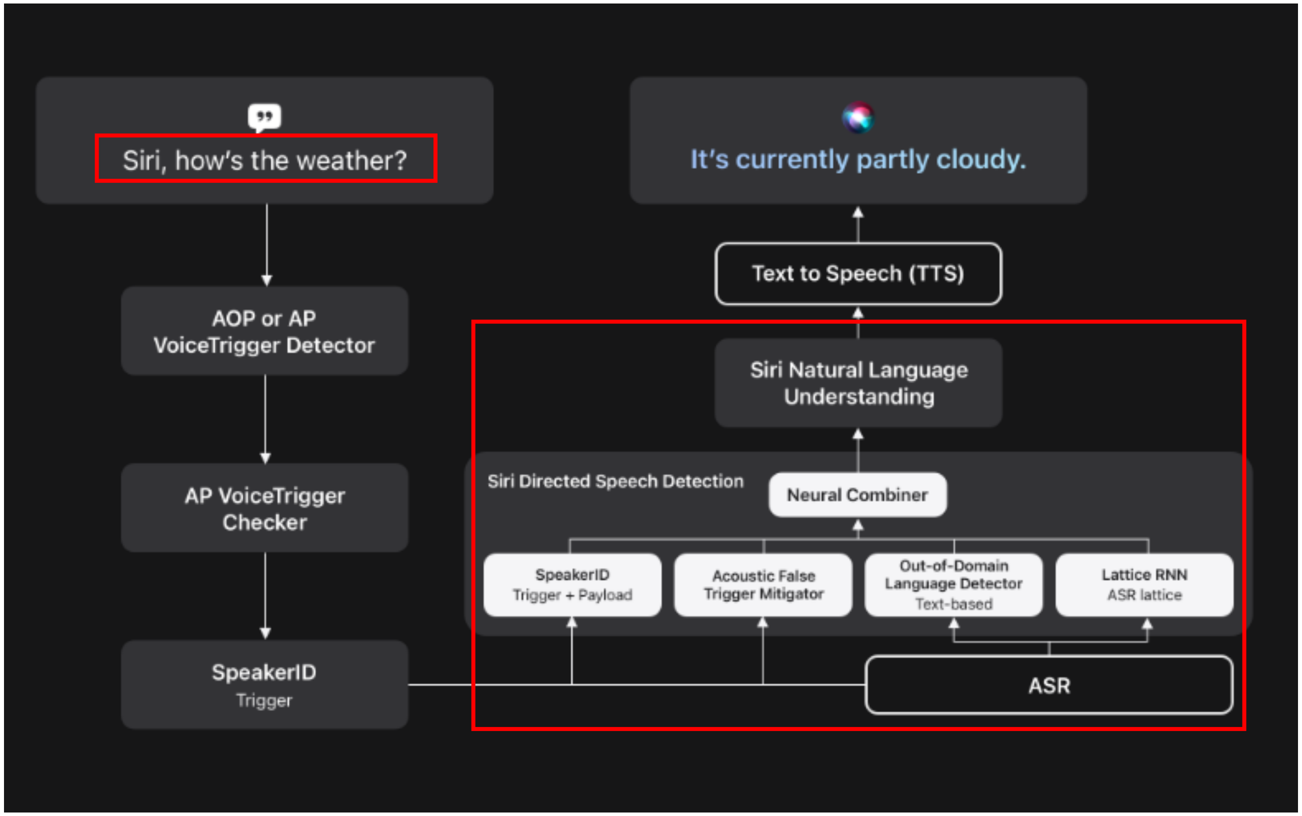


communicating the prepared instructions to a consumer device different from a requesting consumer device

Source: https://developer.apple.com/videos/play/tech-talks/10854/, at 16:45 (annotated)

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[6] The method of claim 1, wherein the information request is a media request and the information is media.	Company performs and/or induces others to perform a method of claim 1, wherein the information request is a media request and the information is media. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri receives an input voice command ("information request") given by the user through their mobile devices. The command comprises a trigger phrase and a subsequent utterance, the trigger phrase being 'Siri' or 'Hey Siri'. Further, Siri responds to the user's voice command and presents an information in the form of text, image, video, and audio ("media") to the user.  Source: https://www.apple.com/siri/ (annotated) Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[7] The method of claim 1, wherein the decoding the information request	Company performs and/or induces others to perform a method of claim 1, wherein the decoding the information request comprises: isolating an utterance from background noise. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri uses automatic speech recognition and natural language processing to process the information request received from the user. It checks whether the user command is directed towards Siri or not and then identifies the intent of the command using a Siri Directed Speech Detection (SDSD) system. The SDSD comprises various False Trigger

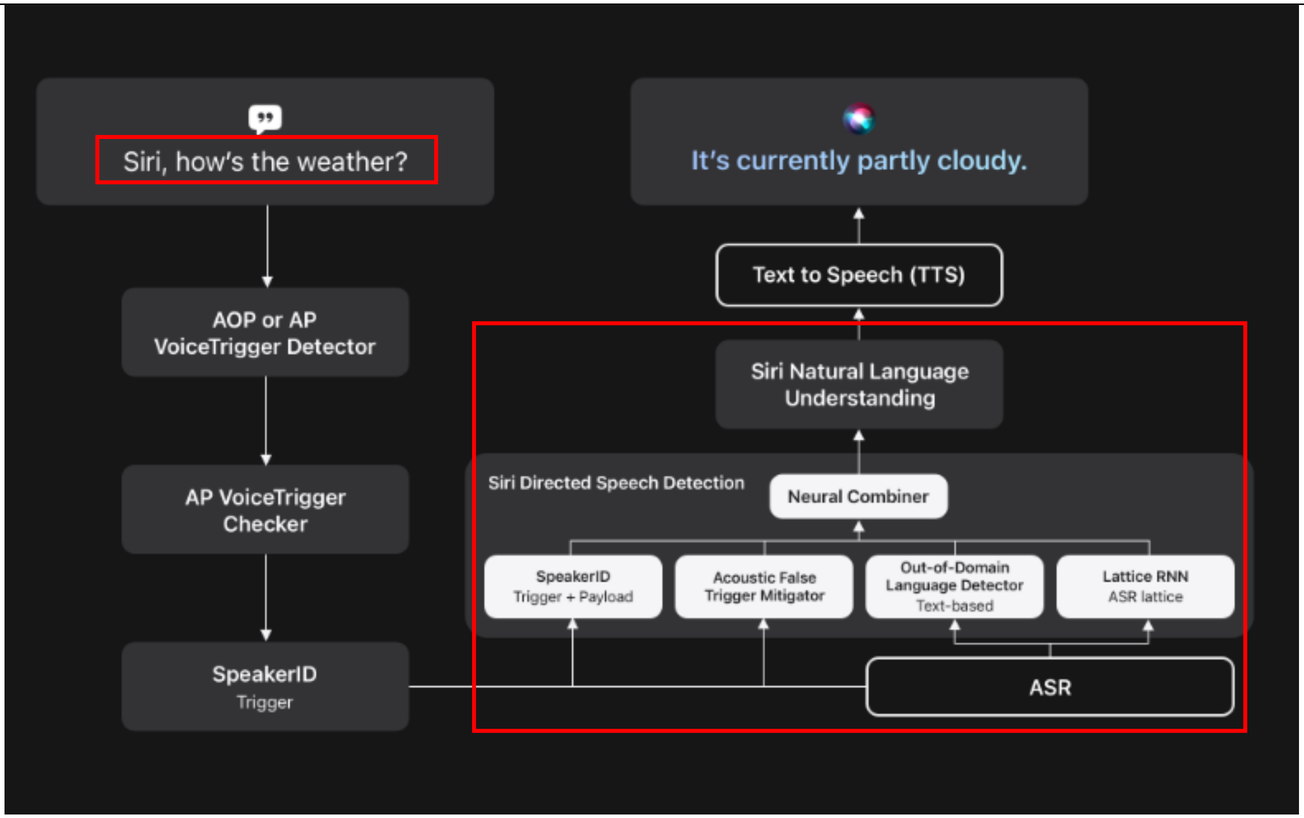
comprises: isolating an utterance from background noise.	Mitigation (FTM) systems that differentiates the true triggers ("utterance") from the false triggers such as background noise in the input voice command.  Source: https://machinelearning.apple.com/research/voice-trigger#figure1

	Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled). Source: https://machinelearning.apple.com/research/hey-siri When a voice trigger detection mechanism detects a trigger, the system starts processing user audio using a full-blown ASR system. A dedicated algorithm determines the end-of-speech event, at which point we obtain the ASR output and the decoding lattice. We use word-aligned lattices such that each arc corresponds to Source: https://machinelearning.apple.com/research/voice-trigger model scores, text, etc. NLU signals are comprised of domain classification features such as domain categories, domain scores, sequence labels of the user request transcription, etc. An intent is a combination of ASR and NLU signals. We refer to these signals as *understanding signals* decoded by ASR and NLU sub-systems. Every intent is encoded into a vector space and this process is described in Section 4.1. Our task is to produce a ranked list of intents using information-state in addition to understanding signals to choose the best response. Source: https://arxiv.org/pdf/2005.00119.pdf, Page 2

	False Trigger Mitigation (FTM) Although the trigger-phrase detection algorithms are precise and reliable, the operating point may allow nontrigger speech or background noise to unexpectedly falsely trigger the device, despite the user not having spoken the trigger phrase, according to the paper Streaming Transformer for Hardware Efficient Voice Trigger Detection and False Trigger Mitigation. ↗ To minimize false triggers, we implement an additional trigger phrase detector that utilizes a significantly larger statistical model. This detector would analyze the complete utterance, allowing for a more precise audio analysis and the ability to override the device's initial trigger decision. We call this the Siri directed speech detection (SDSD) system. We deploy three distinct types of FTM systems to reduce the voice trigger system from responding to unintended false triggers. Each system tries to leverage different clues to identify false triggers. Source: https://machinelearning.apple.com/research/voice-trigger

	In this article, we will discuss how Apple has designed a high-accuracy, privacy-centric, power-efficient, on-device voice trigger system with multiple stages to enable natural voice-driven interactions with Apple devices. The voice trigger system supports several Apple device categories like iPhone, iPad, HomePod, AirPods, Mac, Apple Watch, and Apple Vision Pro. Apple devices simultaneously support two keywords for voice trigger detection: "Hey Siri" and "Siri." We address four specific challenges of voice trigger detection in this article: • Distinguishing a device's primary user from other speakers • Identifying and rejecting false triggers from background noise • Identifying and rejecting acoustic segments that are phonetically similar to trigger phrases • Supporting a shorter phonetically challenging trigger phrase ("Siri") across multiple locales Source: https://machinelearning.apple.com/research/voice-trigger

	We do not rely on the one-best ASR hypothesis for FTM because the acoustic and language models can sometimes "hallucinate" the trigger-phrase. Instead, our approach leverages the whole ASR lattice for FTM. Along with the trigger phrase audio, we expect to exploit the uncertainty in the post-trigger-phrase audio as well. True triggers typically have device-directed speech (for example, "Siri, what time is it?") with limited vocabulary and query-like grammar, whereas false triggers may have random noise or background speech (for example, "Let's go grab lunch"). The decoding lattices explicitly exhibit these differences, and we model them using LSTM-based RNNs. background noise Source: https://machinelearning.apple.com/research/voice-trigger (annotated) Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[8] The method of claim 1, wherein the decoding the information request comprises:	Company performs and/or induces others to perform a method of claim 1, which includes decoding the information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri uses automatic speech recognition and natural language processing to process the information request received from the user. It checks whether the user command is directed towards Siri or not and then identifies the intent of the command using a Siri Directed Speech Detection (SDSD) system. The SDSD comprises various False Trigger Mitigation (FTM) systems such as Acoustic FTM, Out-of-domain Language Detector, and Lattice RNN which decode the input command and convert it into the intent.

	 Source: https://machinelearning.apple.com/research/voice-trigger#figure1

	Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled). Source: https://machinelearning.apple.com/research/hey-siri When a voice trigger detection mechanism detects a trigger, the system starts processing user audio using a full-blown ASR system. A dedicated algorithm determines the end-of-speech event, at which point we obtain the ASR output and the decoding lattice. We use word-aligned lattices such that each arc corresponds to Source: https://machinelearning.apple.com/research/voice-trigger model scores, text, etc. NLU signals are comprised of domain classification features such as domain categories, domain scores, sequence labels of the user request transcription, etc. An intent is a combination of ASR and NLU signals. We refer to these signals as *understanding signals* decoded by ASR and NLU sub-systems. Every intent is encoded into a vector space and this process is described in Section 4.1. Our task is to produce a ranked list of intents using information-state in addition to understanding signals to choose the best response. Source: https://arxiv.org/pdf/2005.00119.pdf, Page 2 Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[8.1] determining a background noise portion of the information request; and	Company performs and/or induces others to perform a step of claim 1, wherein the decoding the information request comprises: determining a background noise portion of the information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, to minimize false triggers, Siri utilizes Siri directed speech detection (SDSD) system that analyzes the complete utterance, allowing for a more precise audio analysis where multiple FTM systems are used to identify false triggers. Therefore, it would be apparent to a person having ordinary skill in the art that the background noise portion is determined. In this article, we will discuss how Apple has designed a high-accuracy, privacy-centric, power-efficient, on-device voice trigger system with multiple stages to enable natural voice-driven interactions with Apple devices. The voice trigger system supports several Apple device categories like iPhone, iPad, HomePod, AirPods, Mac, Apple Watch, and Apple Vision Pro. Apple devices simultaneously support two keywords for voice trigger detection: “Hey Siri” and “Siri.” We address four specific challenges of voice trigger detection in this article: • Distinguishing a device’s primary user from other speakers • Identifying and rejecting false triggers from background noise • Identifying and rejecting acoustic segments that are phonetically similar to trigger phrases • Supporting a shorter phonetically challenging trigger phrase (“Siri”) across multiple locales Source: https://machinelearning.apple.com/research/voice-trigger

	False Trigger Mitigation (FTM) Although the trigger-phrase detection algorithms are precise and reliable, the operating point may allow nontrigger speech or background noise to unexpectedly falsely trigger the device, despite the user not having spoken the trigger phrase, according to the paper Streaming Transformer for Hardware Efficient Voice Trigger Detection and False Trigger Mitigation. ↗ To minimize false triggers, we implement an additional trigger phrase detector that utilizes a significantly larger statistical model. This detector would analyze the complete utterance, allowing for a more precise audio analysis and the ability to override the device's initial trigger decision. We call this the Siri directed speech detection (SDSD) system. We deploy three distinct types of FTM systems to reduce the voice trigger system from responding to unintended false triggers. Each system tries to leverage different clues to identify false triggers. Source: https://machinelearning.apple.com/research/voice-trigger Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[8.2] subtracting the background noise portion from the information request.	Company performs and/or induces others to perform a step of claim 1, wherein the decoding the information request comprises: subtracting the background noise portion from the information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, to minimize false triggers, Siri utilizes Siri directed speech detection (SDSD) system that analyzes the complete utterance, allowing for a more precise audio analysis where multiple FTM systems are used to identify false triggers. Therefore, it would be apparent to a person having ordinary skill in the art that the background noise portion is subtracted from the information request.

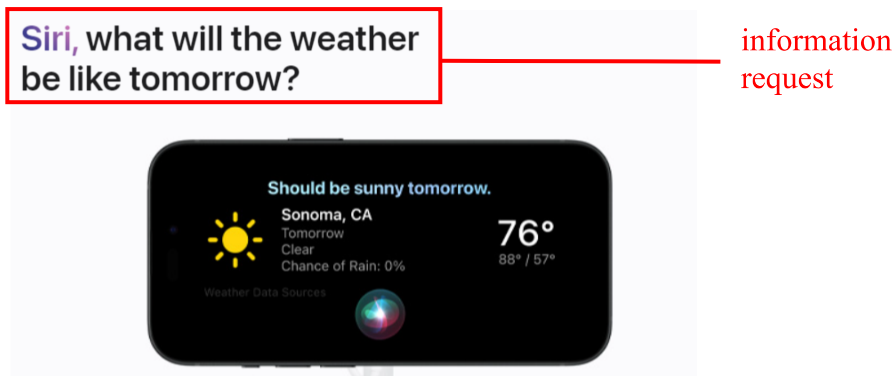
	In this article, we will discuss how Apple has designed a high-accuracy, privacy-centric, power-efficient, on-device voice trigger system with multiple stages to enable natural voice-driven interactions with Apple devices. The voice trigger system supports several Apple device categories like iPhone, iPad, HomePod, AirPods, Mac, Apple Watch, and Apple Vision Pro. Apple devices simultaneously support two keywords for voice trigger detection: “Hey Siri” and “Siri.” We address four specific challenges of voice trigger detection in this article: • Distinguishing a device’s primary user from other speakers • Identifying and rejecting false triggers from background noise • Identifying and rejecting acoustic segments that are phonetically similar to trigger phrases • Supporting a shorter phonetically challenging trigger phrase (“Siri”) across multiple locales Source: https://machinelearning.apple.com/research/voice-trigger

	False Trigger Mitigation (FTM) Although the trigger-phrase detection algorithms are precise and reliable, the operating point may allow nontrigger speech or background noise to unexpectedly falsely trigger the device, despite the user not having spoken the trigger phrase, according to the paper Streaming Transformer for Hardware Efficient Voice Trigger Detection and False Trigger Mitigation. ↗ To minimize false triggers, we implement an additional trigger phrase detector that utilizes a significantly larger statistical model. This detector would analyze the complete utterance, allowing for a more precise audio analysis and the ability to override the device's initial trigger decision. We call this the Siri directed speech detection (SDSD) system. We deploy three distinct types of FTM systems to reduce the voice trigger system from responding to unintended false triggers. Each system tries to leverage different clues to identify false triggers. Source: https://machinelearning.apple.com/research/voice-trigger Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[9] The method of claim 1, wherein the receiving an information request comprises:	Company performs and/or induces others to perform a method of claim 1, which includes receiving an information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri receives an input voice command (“information request”) given by the user through their mobile devices. The command comprises a trigger phrase and a subsequent utterance, the trigger phrase being ‘Siri’ or ‘Hey Siri’.

<table>
<tr>
<td></td>
<td>
Use Siri on all your Apple devices

Use Siri to help you with the things you need to find, know or do every day. Use your voice or press a button to get Siri’s attention, then say what you need. Locate your Apple device below to find out how to use Siri.

Source: https://support.apple.com/en-us/105020



Source: https://www.apple.com/siri/ (annotated)

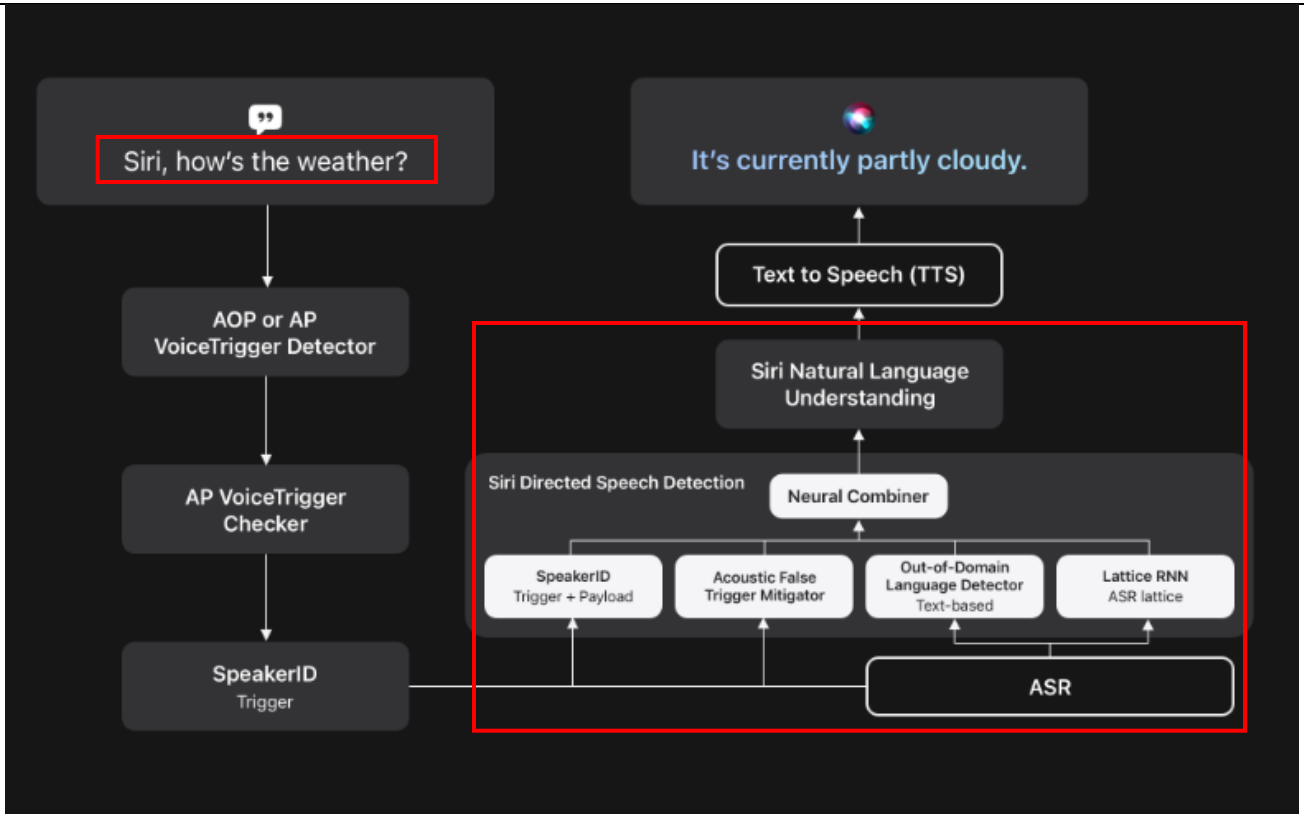
Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
</td>
</tr>
<tr>
<td>[9.1] receiving an information request including at least an audio portion during which a speaker is</td>
<td>
Company performs and/or induces others to perform a step of claim 1, wherein the receiving an information request comprises: receiving an information request including at least an audio portion during which a speaker is silent, the audio portion during which a speaker is silent representing a background noise portion.

This element is infringed literally, or in the alternative, under the doctrine of equivalents.

For example, Siri receives the user’s voice input, that is further converted into a sequence of frames which are then fed to a Deep Neural Network (DNN) model for training using probability distribution over speech sound classes. These classes include trigger phrase phonemes, silence and background speech. Therefore, it would be apparent to a person
</td>
</tr>
</table>

silent, the audio portion during which a speaker is silent representing a background noise portion.	having ordinary skill in the art that Siri receives audio portion during which a speaker is silent. Furthermore, the silent part of the speech is not considered as the trigger phrase phonemes and is therefore, a part of background noise. We used a corpus of speech to train the DNN for which the main Siri recognizer provided a sound class label for each frame. There are thousands of sound classes used by the main recognizer, but only about twenty are needed to account for the target phrase (including an initial silence), and one large class class for everything else. The training process attempts to produce DNN outputs approaching 1 for frames that are labelled with the relevant states and phones, based only on the local sound pattern. The training process adjusts the weights using standard back-propagation and stochastic gradient descent. We have used a variety of neural network training software toolkits, including Theano, Tensorflow, and Kaldi. Source: https://machinelearning.apple.com/research/hey-siri The next strip up (with the yellow diagonal) shows the output of the acoustic model. At each frame there is one output for each position in the phrase, plus others for silence and other speech sounds. The final score, shown at the top, is obtained by adding up the local scores along the bright diagonal according to Equation 1. Note that the score rises to a peak just after the whole phrase enters the system. Source: https://machinelearning.apple.com/research/hey-siri Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[10] The method of claim 9, wherein the decoding the information request comprises:	Company performs and/or induces others to perform a method of claim 9, which includes decoding the information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri uses automatic speech recognition and natural language processing to process the information request received from the user. It checks whether the user command is directed towards Siri or not and then identifies the intent of the command using a Siri Directed Speech Detection (SDSD) system. The SDSD comprises various False Trigger Mitigation (FTM) systems such as Acoustic FTM, Out-of-domain Language Detector, and Lattice RNN which decode the input command and convert it into the intent.

	 Source: https://machinelearning.apple.com/research/voice-trigger#figure1 (annotated)

	Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled). Source: https://machinelearning.apple.com/research/hey-siri When a voice trigger detection mechanism detects a trigger, the system starts processing user audio using a full-blown ASR system. A dedicated algorithm determines the end-of-speech event, at which point we obtain the ASR output and the decoding lattice. We use word-aligned lattices such that each arc corresponds to Source: https://machinelearning.apple.com/research/voice-trigger model scores, text, etc. NLU signals are comprised of domain classification features such as domain categories, domain scores, sequence labels of the user request transcription, etc. An intent is a combination of ASR and NLU signals. We refer to these signals as *understanding signals* decoded by ASR and NLU sub-systems. Every intent is encoded into a vector space and this process is described in Section 4.1. Our task is to produce a ranked list of intents using information-state in addition to understanding signals to choose the best response. Source: https://arxiv.org/pdf/2005.00119.pdf, Page 2 Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[10.1] using the background noise portion to filter background noise from at least an audio portion of the information request including an utterance.	Company performs and/or induces others to perform a step of claim 9, wherein the decoding the information request comprises: using the background noise portion to filter background noise from at least an audio portion of the information request including an utterance. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri utilizes Siri directed speech detection (SDSD) system that analyzes the complete utterance, allowing for a more precise audio analysis where multiple FTM systems are used to identify false triggers. Therefore, it would be apparent to a person having ordinary skill in the art that Siri uses background noise portion to filter background noise from the information request including an utterance. We used a corpus of speech to train the DNN for which the main Siri recognizer provided a sound class label for each frame. There are thousands of sound classes used by the main recognizer, but only about twenty are needed to account for the target phrase (including an initial silence), and one large class class for everything else. The training process attempts to produce DNN outputs approaching 1 for frames that are labelled with the relevant states and phones, based only on the local sound pattern. The training process adjusts the weights using standard back-propagation and stochastic gradient descent. We have used a variety of neural network training software toolkits, including Theano, Tensorflow, and Kaldi. Source: https://machinelearning.apple.com/research/hey-siri The next strip up (with the yellow diagonal) shows the output of the acoustic model. At each frame there is one output for each position in the phrase, plus others for silence and other speech sounds. The final score, shown at the top, is obtained by adding up the local scores along the bright diagonal according to Equation 1. Note that the score rises to a peak just after the whole phrase enters the system.

	Source: https://machinelearning.apple.com/research/hey-siri In this article, we will discuss how Apple has designed a high-accuracy, privacy-centric, power-efficient, on-device voice trigger system with multiple stages to enable natural voice-driven interactions with Apple devices. The voice trigger system supports several Apple device categories like iPhone, iPad, HomePod, AirPods, Mac, Apple Watch, and Apple Vision Pro. Apple devices simultaneously support two keywords for voice trigger detection: “Hey Siri” and “Siri.” We address four specific challenges of voice trigger detection in this article: • Distinguishing a device’s primary user from other speakers • Identifying and rejecting false triggers from background noise • Identifying and rejecting acoustic segments that are phonetically similar to trigger phrases • Supporting a shorter phonetically challenging trigger phrase (“Siri”) across multiple locales Source: https://machinelearning.apple.com/research/voice-trigger

False Trigger Mitigation (FTM)

Although the trigger-phrase detection algorithms are precise and reliable, the operating point may allow nontrigger speech or background noise to unexpectedly falsely trigger the device, despite the user not having spoken the trigger phrase, according to the paper Streaming Transformer for Hardware Efficient Voice Trigger Detection and False Trigger Mitigation. ↗ To minimize false triggers, we implement an additional trigger phrase detector that utilizes a significantly larger statistical model. This detector would analyze the complete utterance, allowing for a more precise audio analysis and the ability to override the device's initial trigger decision. We call this the Siri directed speech detection (SDSD) system. We deploy three distinct types of FTM systems to reduce the voice trigger system from responding to unintended false triggers. Each system tries to leverage different clues to identify false triggers.

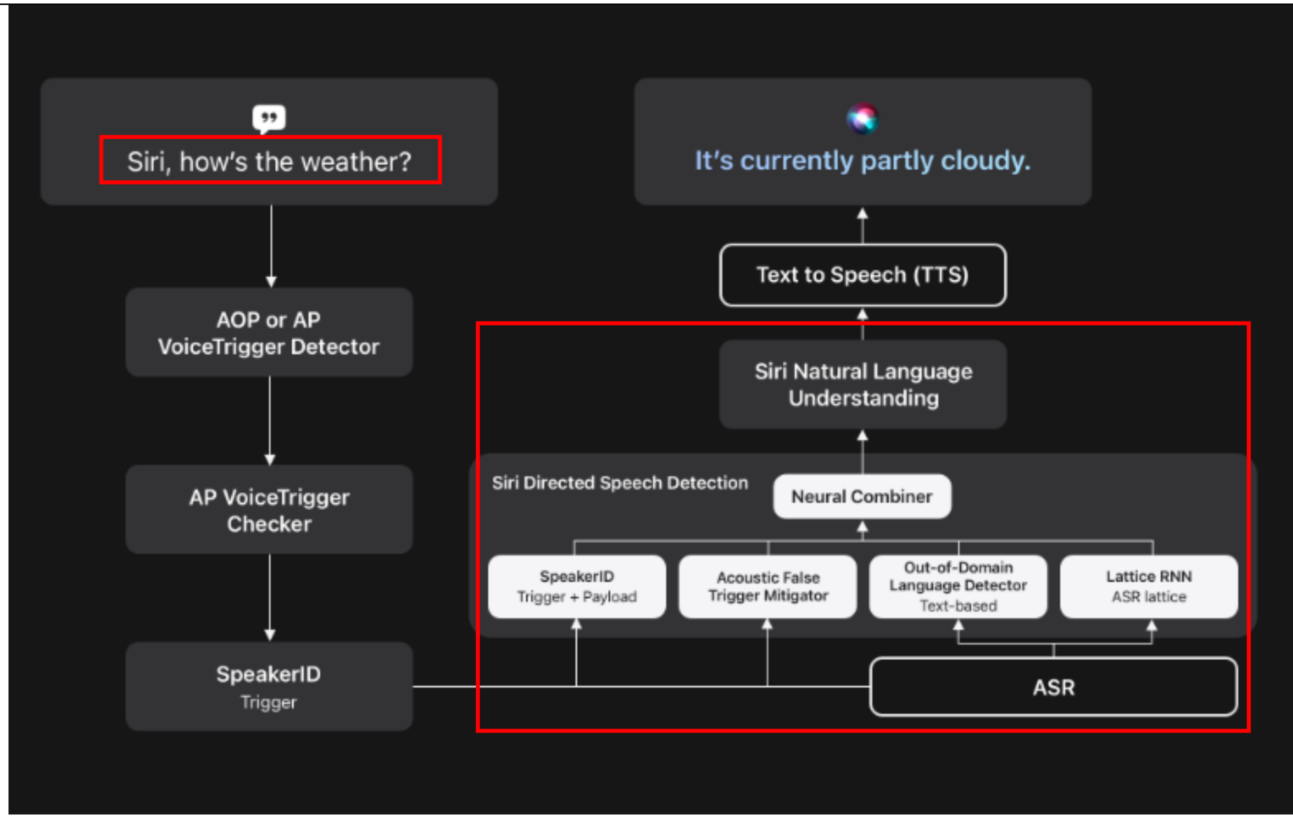
Source: https://machinelearning.apple.com/research/voice-trigger

We do not rely on the one-best ASR hypothesis for FTM because the acoustic and language models can sometimes “hallucinate” the trigger-phrase. Instead, our approach leverages the whole ASR lattice for FTM. Along with the trigger phrase audio, we expect to exploit the uncertainty in the post-trigger-phrase audio as well. True triggers typically have device-directed speech (for example, “Siri, what time is it?”) with limited vocabulary and query-like grammar, whereas false triggers may have random noise or background speech (for example, “Let’s go grab lunch”). The decoding lattices explicitly exhibit these differences, and we model them using LSTM-based RNNs.

background noise

Source: https://machinelearning.apple.com/research/voice-trigger (annotated)

	Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[13] The method of claim 1, wherein the decoding the information request comprises: applying one or more of a pitch-shift or a time-shift to the information request.	Company performs and/or induces others to perform a method of claim 1, wherein the decoding the information request comprises: applying one or more of a pitch-shift or a time-shift to the information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri uses automatic speech recognition and natural language processing to process the information request received from the user. Further, the received voice input that contains the keywords ‘Siri’ or ‘Hey Siri’ is analyzed using a ‘Hey Siri’ detector where the acoustic pattern of the voice is converted into a probability distribution of multiple speech frames using DNN and the temporal integration process (“applying one or more of a pitch-shift or a time-shift to the information request”) is used to compute a confidence score such that the Siri wakes up.



Source: https://machinelearning.apple.com/research/voice-trigger#figure1 (annotated)

When a voice trigger detection mechanism detects a trigger, the system starts processing user audio using a full-blown ASR system. A dedicated algorithm determines the end-of-speech event, at which point we obtain the ASR output and the decoding lattice. We use word-aligned lattices such that each arc corresponds to

Source: https://machinelearning.apple.com/research/voice-trigger

	model scores, text, etc. NLU signals are comprised of domain classification features such as domain categories, domain scores, sequence labels of the user request transcription, etc. An intent is a combination of ASR and NLU signals. We refer to these signals as *understanding signals* decoded by ASR and NLU sub-systems. Every intent is encoded into a vector space and this process is described in Section 4.1. Our task is to produce a ranked list of intents using information-state in addition to understanding signals to choose the best response. Source: https://arxiv.org/pdf/2005.00119.pdf, Page 2 The "Hey Siri" feature allows users to invoke Siri hands-free. A very small speech recognizer runs all the time and listens for just those two words. When it detects "Hey Siri", the rest of Siri parses the following speech as a command or query. The "Hey Siri" detector uses a Deep Neural Network (DNN) to convert the acoustic pattern of your voice at each instant into a probability distribution over speech sounds. It then uses a temporal integration process to compute a confidence score that the phrase you uttered was "Hey Siri". If the score is high enough, Siri wakes up. This article takes a look at the underlying technology. It is aimed primarily at readers who know something of machine learning but less about speech recognition. Source: https://machinelearning.apple.com/research/hey-siri The microphone in an iPhone or Apple Watch turns your voice into a stream of instantaneous waveform samples, at a rate of 16000 per second. A spectrum analysis stage converts the waveform sample stream to a sequence of frames, each describing the sound spectrum of approximately 0.01 sec. About twenty of these frames at a time (0.2 sec of audio) are fed to the acoustic model, a Deep Neural Network (DNN) which converts each of these acoustic patterns into a probability distribution over a set of speech sound classes: those used in the "Hey Siri" phrase, plus silence and other speech, for a total of about 20 sound classes. See Figure 2.

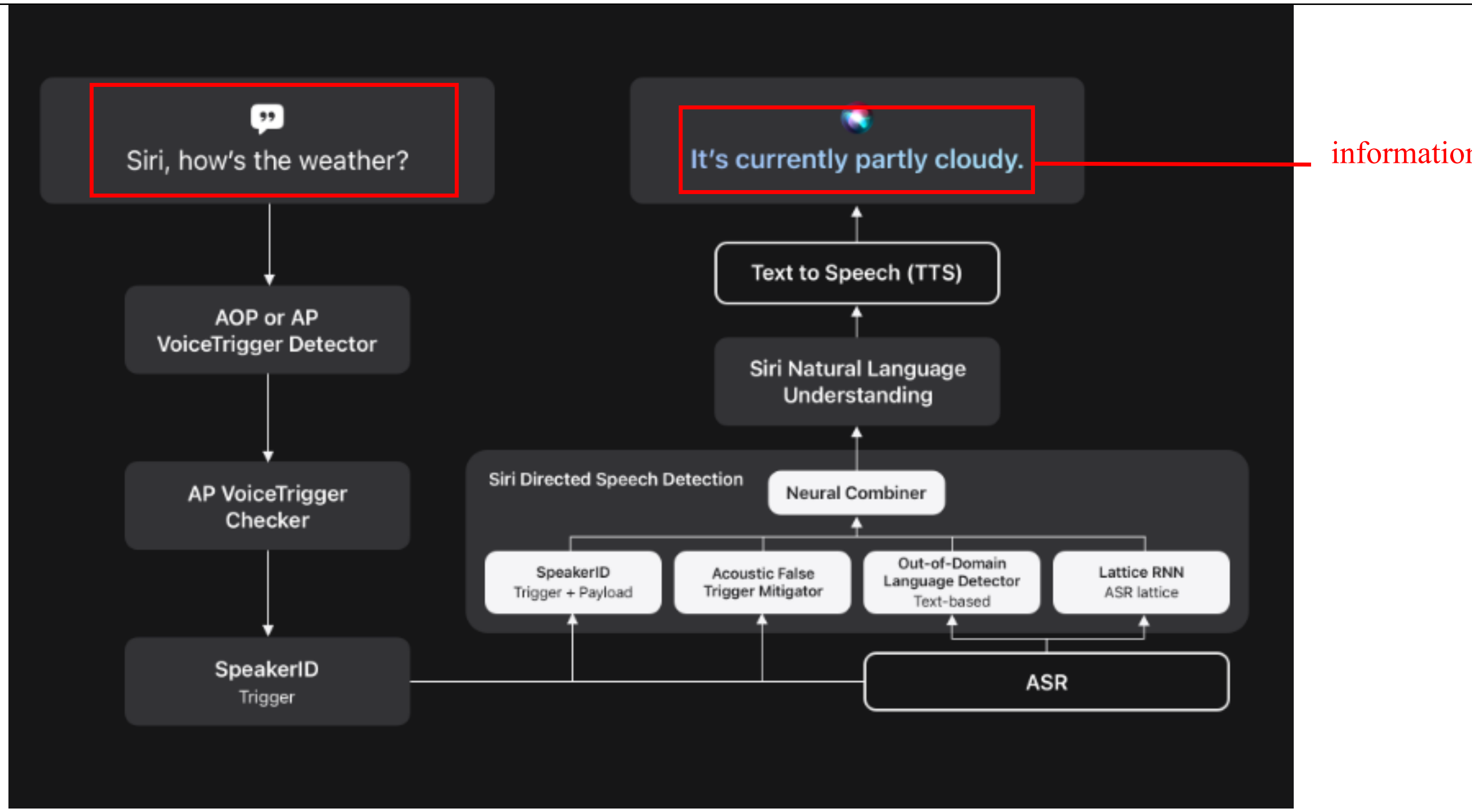
	Source: https://machinelearning.apple.com/research/hey-siri The first stage in the voice trigger detection system is a low-power, first-pass detector that receives streaming input from the microphone and is a deep neural network (DNN) hidden markov model (HMM) based keyword spotting model, as discussed in our research article, Personalized Hey Siri. › The DNN predicts the state probabilities of a given speech frame. At the same time, the HMM decoder uses dynamic programming to combine the DNN predictions of multiple speech frames to compute the keyword detection score. The DNN output contains 23 states: • 21 corresponding to seven phonemes of the trigger phrases (three states for each phoneme) • One state for silence • One for background Source: https://machinelearning.apple.com/research/voice-trigger Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[29.P] A system for providing information discovery and retrieval, the system comprising	Apple ("Company") makes, uses, sells and/or offers to sell a system for providing information discovery and retrieval. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Company provides Siri, an intelligent voice assistant that receives voice commands from a user through a mobile device such as an iPhone and retrieves information ("information discovery and retrieval") related to the voice command.

SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

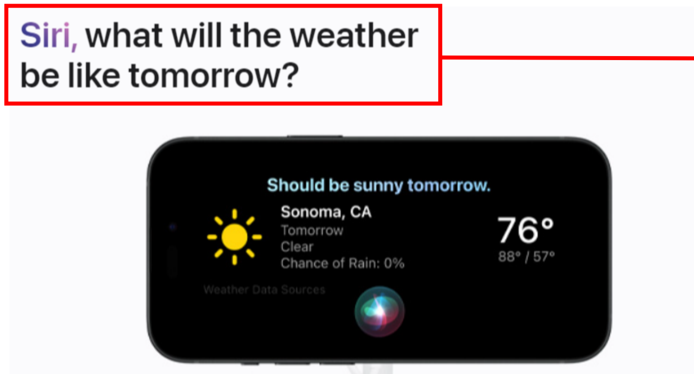
information discovery and retrieval

Source: https://developer.apple.com/design/human-interface-guidelines/siri (annotated)

	 Source: https://machinelearning.apple.com/research/voice-trigger#figure1 (annotated) Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[29.1] a processor module, the processor module	Company provides a system for providing information discovery and retrieval, the system comprising: a processor module. This element is infringed literally, or in the alternative, under the doctrine of equivalents.

configured at least for performing the steps of:	For example, Siri processes the user voice input using an Always on Processor (AOP). It analyses the voice input to identify the trigger phrase. Upon detection of the trigger phrase by AOP, it uses the Application Processor (AP) having a high-precision voice trigger checker system. The multistage architecture for the voice trigger system is shown in Figure 1. On mobile devices, audio is analyzed in a streaming fashion on the Always On Processor (AOP). An on-device ring buffer is used to store this streaming audio. The user's input audio is then analyzed by a streaming high-recall voice trigger detector system, and any audio that does not contain the trigger keywords is discarded. Audio that may contain the trigger keywords is analyzed using a high-precision voice trigger checker system on the Application Processor (AP). For personal devices, like iPhone, the speaker identification (speakerID) system is used to analyze if the trigger phrase is uttered by the owner of the device or another user. Siri directed speech detection (SDSD) analyzes the full user utterance, including the trigger phrase segment, and decides whether to mitigate any potential false voice trigger utterances. We detail individual systems in the following sections. Source: https://machinelearning.apple.com/research/voice-trigger Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[29.2] receiving an information request from a consumer device over a communications network;	Company provides a system for providing information discovery and retrieval, the system comprising: a processor module, the processor module configured at least for performing the steps of: receiving an information request from a consumer device over a communications network. This element is infringed literally, or in the alternative, under the doctrine of equivalents.

For example, Siri receives an input voice command ("information request") given by the user through their mobile devices ("consumer device") over Internet ("communication network"). The command comprises a trigger phrase and a subsequent utterance, the trigger phrase being 'Siri' or 'Hey Siri'.



information request

Source: https://www.apple.com/siri/ (annotated)

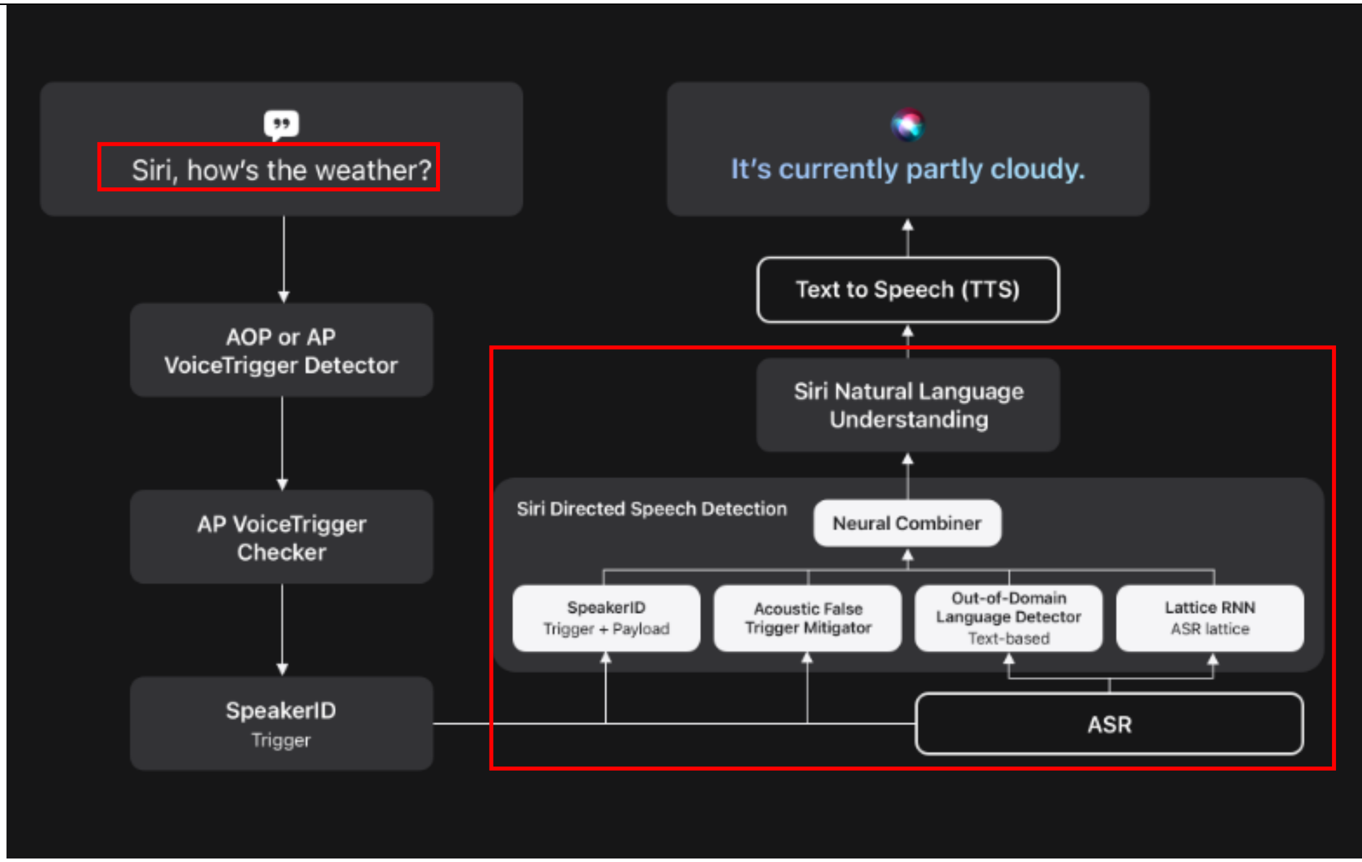
In this article, we will discuss how Apple has designed a high-accuracy, privacy-centric, power-efficient, on-device voice trigger system with multiple stages to enable natural voice-driven interactions with Apple devices. The voice trigger system supports several Apple device categories like iPhone, iPad, HomePod, AirPods, Mac, Apple Watch, and Apple Vision Pro. Apple devices simultaneously support two keywords for voice trigger detection: "Hey Siri" and "Siri."

Source: https://machinelearning.apple.com/research/voice-trigger

Source: https://discussions.apple.com/thread/254229869 (annotated)

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[29.3] decoding the information request;	Company provides a system for providing information discovery and retrieval, the system comprising: a processor module, the processor module configured at least for performing the steps of: decoding the information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri uses automatic speech recognition and natural language processing to process the information request received from the user. It checks whether the user command is directed towards Siri or not and then, identifies the intent of the command using a Siri Directed Speech Detection (SDSD) system. The SDSD comprises various False Trigger Mitigation (FTM) systems such as Acoustic FTM, Out-of-domain Language Detector, and Lattice RNN which decode the input command and convert it into the intent.

	 Source: https://machinelearning.apple.com/research/voice-trigger#figure1

Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled).

Source: https://machinelearning.apple.com/research/hey-siri

When a voice trigger detection mechanism detects a trigger, the system starts processing user audio using a full-blown ASR system. A dedicated algorithm determines the end-of-speech event, at which point we obtain the ASR output and the decoding lattice. We use word-aligned lattices such that each arc corresponds to

Source: https://machinelearning.apple.com/research/voice-trigger

model scores, text, etc. NLU signals are comprised of domain classification features such as domain categories, domain scores, sequence labels of the user request transcription, etc. An intent is a combination of ASR and NLU signals. We refer to these signals as *understanding signals* decoded by ASR and NLU sub-systems. Every intent is encoded into a vector space and this process is described in Section 4.1. Our task is to produce a ranked list of intents using information-state in addition to understanding signals to choose the best response.

Source: https://arxiv.org/pdf/2005.00119.pdf, Page 2

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[29.4] discovering information using the decoded information request;	Company provides a system for providing information discovery and retrieval, the system comprising: a processor module, the processor module configured at least for performing the steps of: discovering information using the decoded information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, after the intents ("decoded information request") are determined, the relevant information is retrieved based on the intent.

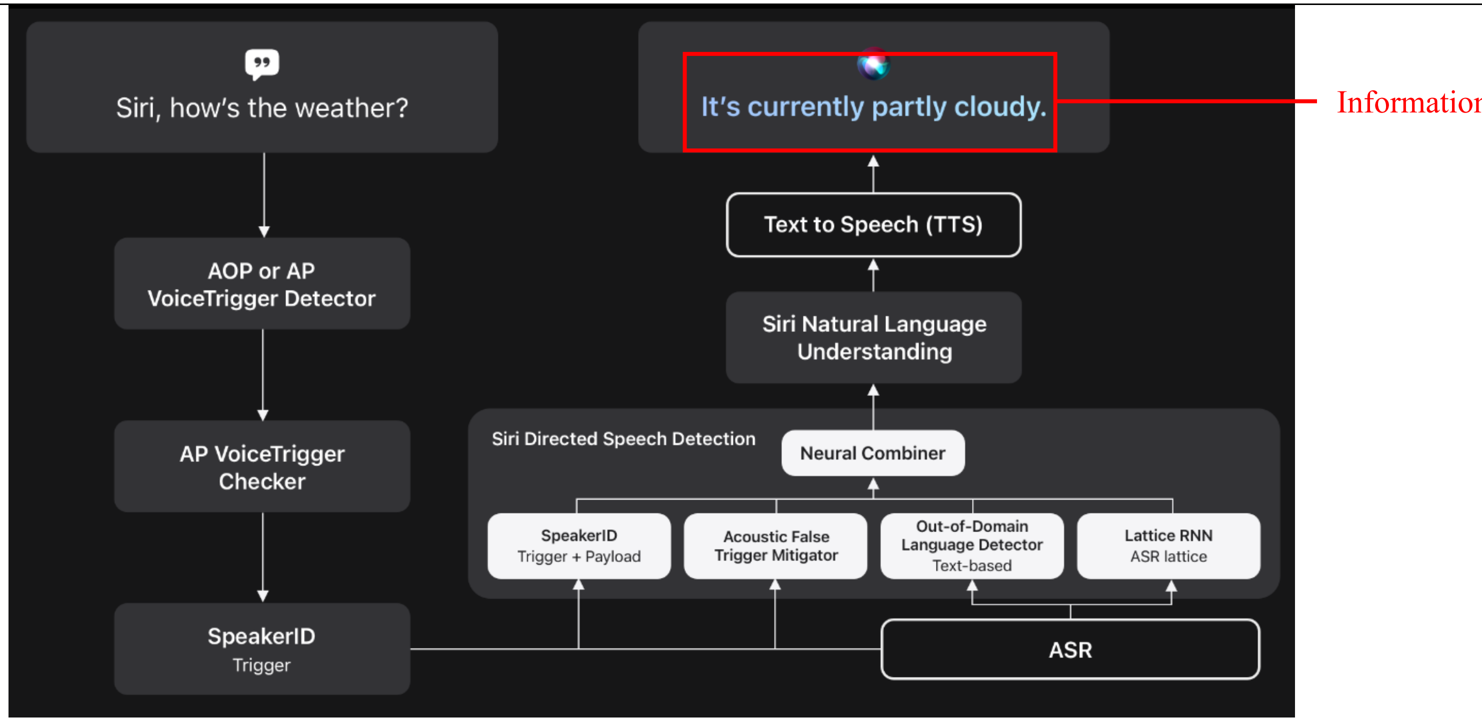
SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

discovering information

Source: https://developer.apple.com/design/human-interface-guidelines/siri (annotated)

	Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled). Source: https://machinelearning.apple.com/research/hey-siri

	 Source: https://machinelearning.apple.coms/research/voice-trigger (annotated) Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[29.5] preparing instructions for accessing the information, the	Company provides a system for providing information discovery and retrieval, the system comprising: a processor module, the processor module configured at least for performing the steps of: preparing instructions for accessing the information. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, after converting audio requests to the intents, Siri provides conversational flow for system intents such that the information is retrieved. These flows help the app to fulfill the user request according to the domain of the intents.

instructions including:	Therefore, it would be apparent to a person having ordinary skill in the art that Siri prepares instructions for accessing the information. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis. Source: https://developer.apple.com/design/human-interface-guidelines/siri/ **System intents** SiriKit defines a large number of system intents that represent common tasks people do, such as playing music, sending messages to friends, and managing notes. For system intents, Siri defines the conversational flow, while your app provides the data to complete the interaction. Source: https://developer.apple.com/design/human-interface-guidelines/siri#System-intents

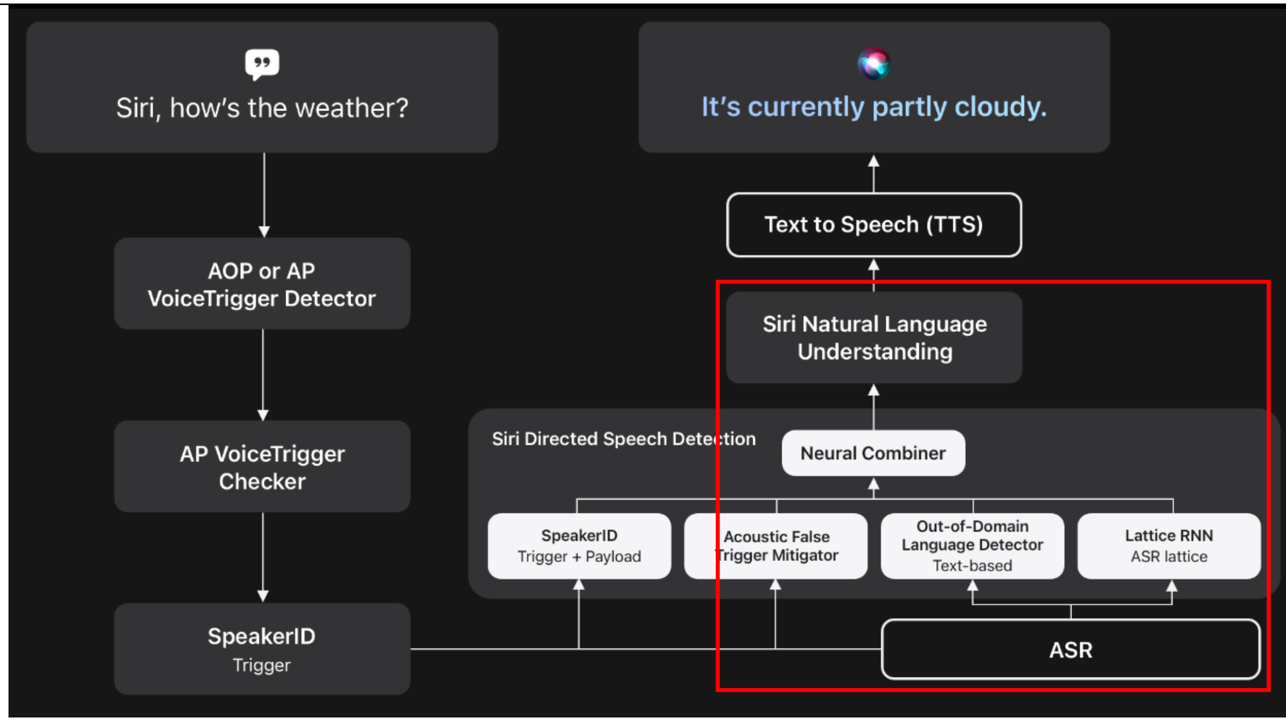
SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

Source: https://developer.apple.com/design/human-interface-guidelines/siri

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[29.6] one or more Automatic Speech Recognition (ASR) grammar codes;	Company provides a system for providing information discovery and retrieval, the system comprising: preparing instructions for accessing the information, the instructions including: one or more Automatic Speech Recognition (ASR) grammar codes. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri does the language processing and semantic analysis to convert the requests into the intents. During semantic analysis, the audio input is matched against a grammar ("one or more Automatic Speech Recognition (ASR) grammar codes") to produce a semantic interpretation of the input. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis. Source: https://developer.apple.com/design/human-interface-guidelines/siri/ **1.4 Semantic Interpretation** A speech recognizer is capable of matching audio input against a grammar to produce a *raw text* transcription (also known as *literal text*) of the detected input. A recognizer may be capable of, but is not required to, perform subsequent processing of the raw text to produce a *semantic interpretation* of the input. Source: https://www.w3.org/TR/2004/REC-speech-grammar-20040316/#S1.3

<table>
<tr>
<td></td>
<td>


Source: https://machinelearning.apple.coms/research/voice-trigger

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
</td>
</tr>
<tr>
<td>[29.7] one or more short text string matching codes; and</td>
<td>
Company provides a system for providing information discovery and retrieval, the system comprising: preparing instructions for accessing the information, the instructions including: one or more short text string matching codes.

This element is infringed literally, or in the alternative, under the doctrine of equivalents.
</td>
</tr>
</table>

For example, Siri does the natural language processing and semantic analysis to convert the requests into the intents, and provides string to match against the data. Since, the relevant information is retrieved according to the intent, upon information and belief, the instructions comprise one or more short text string matching codes.

A closer look at intents

When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis.

Source: https://developer.apple.com/design/human-interface-guidelines/siri/

Overview

Your app likely defines a number of custom types that model the data the app creates or consumes. For example, a music app might define types that represent artists, albums, and tracks. Because those types are unique to your app, the framework can't interpret them until you expose them to system services such as Siri and the Shortcuts app. *Entities* are lightweight types that provide information to the system about your app's data or concepts relating to that data. An entity identifies and queries the data it represents and describes how the system displays that data onscreen.

Source: https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents

	To let people use arbitrary text to find specific entities, adopt the `EntityStringQuery` protocol instead. Queries that adopt this protocol cause the system to display a search field above the list of suggested entities. Implement the required `entities(matching:)` function, and use the provided string to match against your data. For example, a music app might let people search for a specific album by matching against the album name. Source: https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[29.8] one or more information formatting codes operative to format a consumer device display; and	Company provides a system for providing information discovery and retrieval, the system comprising: preparing instructions for accessing the information, the instructions including: one or more information formatting codes operative to format a consumer device display. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, the intents describe how the system displays ("information formatting codes operative to format a consumer device display") the data such as dates, times, and addresses. **Overview** Your app likely defines a number of custom types that model the data the app creates or consumes. For example, a music app might define types that represent artists, albums, and tracks. Because those types are unique to your app, the framework can't interpret them until you expose them to system services such as Siri and the Shortcuts app. *Entities* are lightweight types that provide information to the system about your app's data or concepts relating to that data. An entity identifies and queries the data it represents and describes how the system displays that data onscreen.

	Source: https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents Siri displays entities like dates, times, addresses and currency amounts in a nicely formatted way. This is the result of the application of a process called inverse text normalization (ITN) to the output of a core speech recognition component. To understand the important role ITN plays, consider that, without it, Siri would display "October twenty third twenty sixteen" instead of "October 23, 2016". In this work, we Source: https://machinelearning.apple.com/research/inverse-text-normal Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[29.9] communicating the prepared instructions to the consumer device,	Company provides a system for providing information discovery and retrieval, the system comprising: communicating the prepared instructions to the consumer device. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri communicates the instructions to execute intents to the apps in the user's mobile device such as iPhone such that the relevant information is accessed. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis.

Source: https://developer.apple.com/design/human-interface-guidelines/siri/

SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

Source: https://developer.apple.com/design/human-interface-guidelines/siri

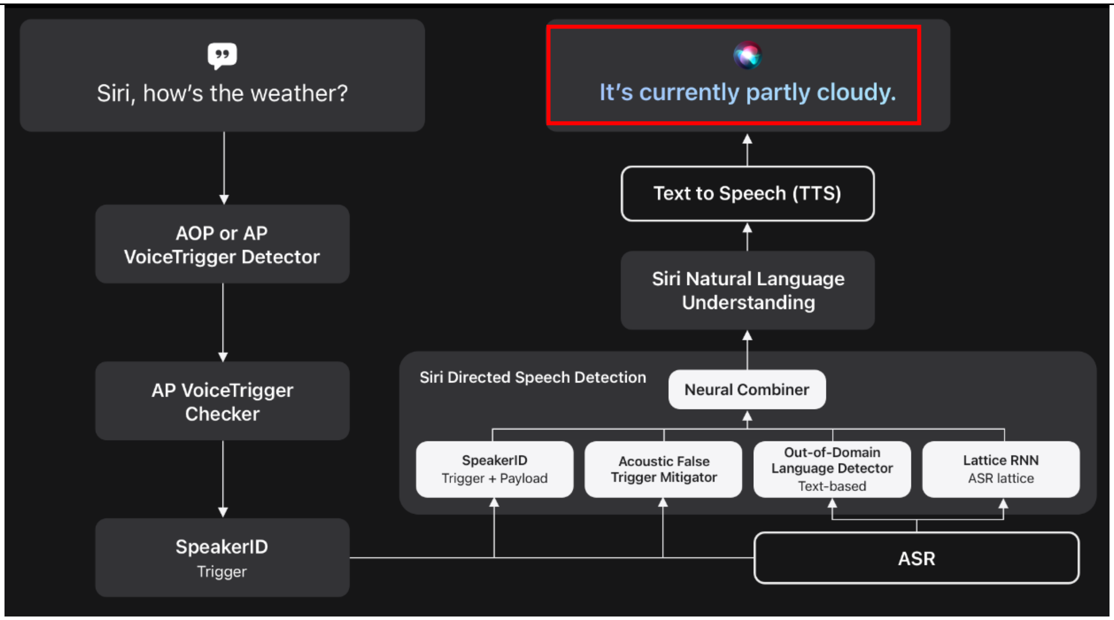
Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

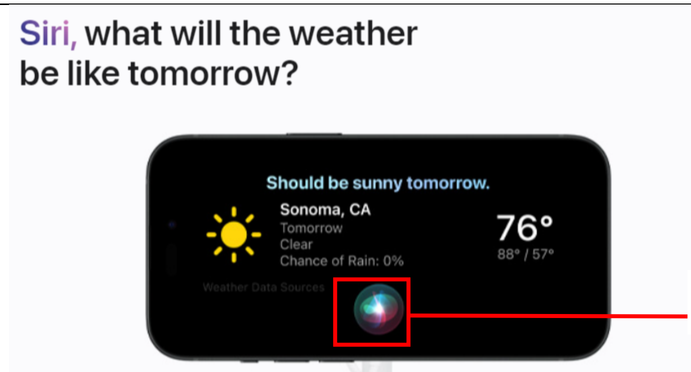
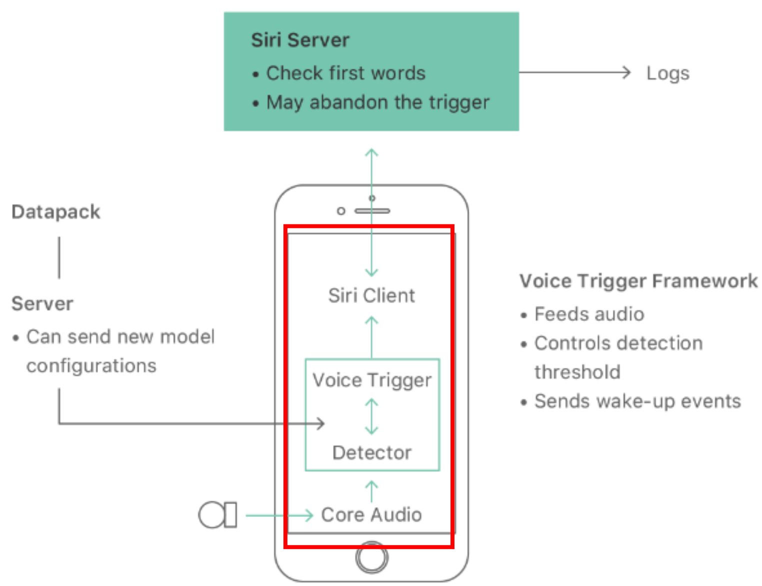
[29.10] wherein the consumer device is configured at least for retrieving the information for presentation using the prepared instructions.	Company provides a system for providing information discovery and retrieval, the system comprising: a processor module wherein the consumer device is configured at least for retrieving the information for presentation using the prepared instructions. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri provides conversational flow for system intents such that the information is retrieved. These flows help the app to fulfill the user request according to the domain of the intents. Upon receiving these instructions, Siri displays ("retrieving the information for presentation") the information from the app on the user's mobile device. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis. Source: https://developer.apple.com/design/human-interface-guidelines/siri/

SiriKit provides the following intents.

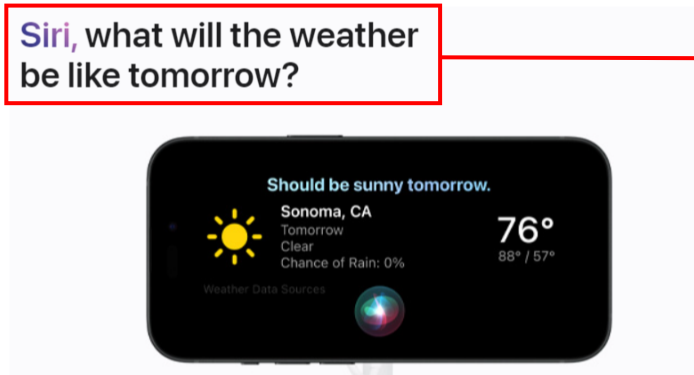
Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

Source: https://developer.apple.com/design/human-interface-guidelines/siri

	 Source: https://machinelearning.apple.com/research/voice-trigger Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[30.P] A computer program product comprising one or more non-transitory computer readable	Apple ("Company") makes, uses, sells and/or offers to sell a computer program product comprising one or more non-transitory computer readable media bearing one or more instructions. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Company provides Siri ("computer program product"), an intelligent voice assistant, compatible with iOS devices such as iPhone. Further, Siri comprises instructions to perform various functionalities such as receiving voice input from a user, retrieves information based on the input, and presents the information to the user. Therefore, it would be apparent to a person having ordinary skill in the art that Siri utilizes memory ("one or more non-transitory computer readable media bearing one or more instructions") of the device to execute these instructions.

media bearing one or more instructions for:	 Source: https://www.apple.com/siri/ (annotated) Figure 1. The Hey Siri flow on iPhone 

	Source: https://machinelearning.apple.com/research/hey-siri **Responsiveness and Power: Two Pass Detection** The "Hey Siri" detector not only has to be accurate, but it needs to be fast and not have a significant effect on battery life. We also need to minimize memory use and processor demand—particularly peak processor demand. To avoid running the main processor all day just to listen for the trigger phrase, the iPhone's Always On Processor (AOP) (a small, low-power auxiliary processor, that is, the embedded Motion Coprocessor) has access to the microphone signal (on 6S and later). We use a small proportion of the AOP's limited processing power to run a detector with a small version of the acoustic model (DNN). When the score exceeds a threshold the motion coprocessor wakes up the main processor, which analyzes the signal using a larger DNN. In the first versions with AOP support, the first detector used a DNN with 5 layers of 32 hidden units and the second detector had 5 layers of 192 hidden units. Source: https://machinelearning.apple.com/research/hey-siri Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[30.1] receiving an information request;	Company provides a computer program product comprising one or more non-transitory computer readable media bearing one or more instructions for: receiving an information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri receives an input voice command ("information request") given by the user through their mobile devices. The command comprises a trigger phase and a subsequent utterance, the trigger phase being 'Siri' or 'Hey Siri'.



information request

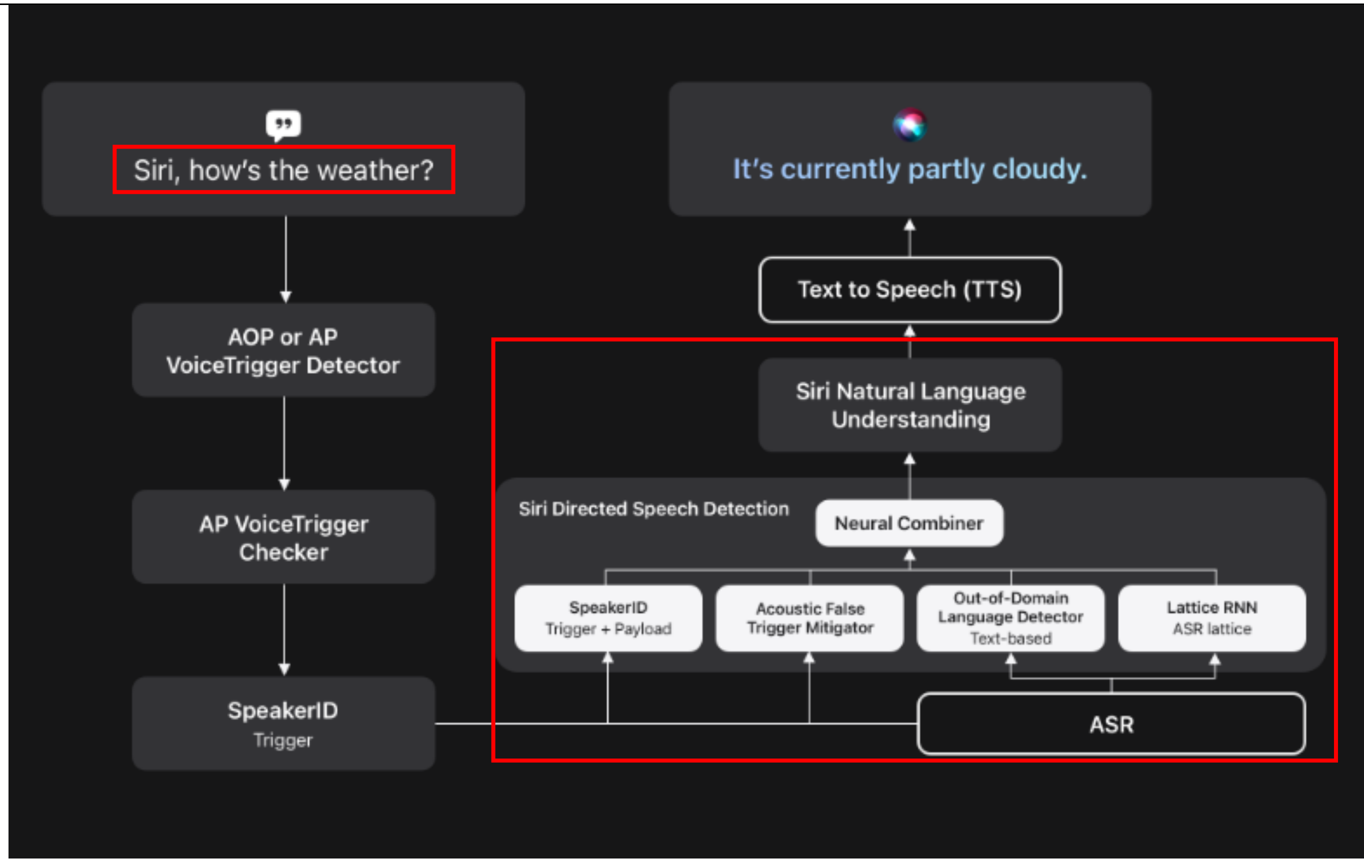
Source: https://www.apple.com/siri/ (annotated)

In this article, we will discuss how Apple has designed a high-accuracy, privacy-centric, power-efficient, on-device voice trigger system with multiple stages to enable natural voice-driven interactions with Apple devices. The voice trigger system supports several Apple device categories like iPhone, iPad, HomePod, AirPods, Mac, Apple Watch, and Apple Vision Pro. Apple devices simultaneously support two keywords for voice trigger detection: "Hey Siri" and "Siri."

Source: https://machinelearning.apple.com/research/voice-trigger

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[30.2] decoding the information request;	Company provides a computer program product comprising one or more non-transitory computer readable media bearing one or more instructions for: decoding the information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri uses automatic speech recognition and natural language processing to process the information request received from the user. It checks whether the user command is directed towards Siri or not and then, identifies the intent of the command using a Siri Directed Speech Detection (SDSD) system. The SDSD comprises various False Trigger Mitigation (FTM) systems such as Acoustic FTM, Out-of-domain Language Detector, and Lattice RNN which decode the input command and convert it into the intent.

	 Source: https://machinelearning.apple.com/research/voice-trigger#figure1

	Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled). Source: https://machinelearning.apple.com/research/hey-siri When a voice trigger detection mechanism detects a trigger, the system starts processing user audio using a full-blown ASR system. A dedicated algorithm determines the end-of-speech event, at which point we obtain the ASR output and the decoding lattice. We use word-aligned lattices such that each arc corresponds to Source: https://machinelearning.apple.com/research/voice-trigger model scores, text, etc. NLU signals are comprised of domain classification features such as domain categories, domain scores, sequence labels of the user request transcription, etc. An intent is a combination of ASR and NLU signals. We refer to these signals as *understanding signals* decoded by ASR and NLU sub-systems. Every intent is encoded into a vector space and this process is described in Section 4.1. Our task is to produce a ranked list of intents using information-state in addition to understanding signals to choose the best response. Source: https://arxiv.org/pdf/2005.00119.pdf, Page 2 Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

[30.3] discovering information using the decoded information request;	Company provides a computer program product comprising one or more non-transitory computer readable media bearing one or more instructions for: discovering information using the decoded information request. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, after the intents ("decoded information request") are determined, the relevant information is retrieved based on the intent.

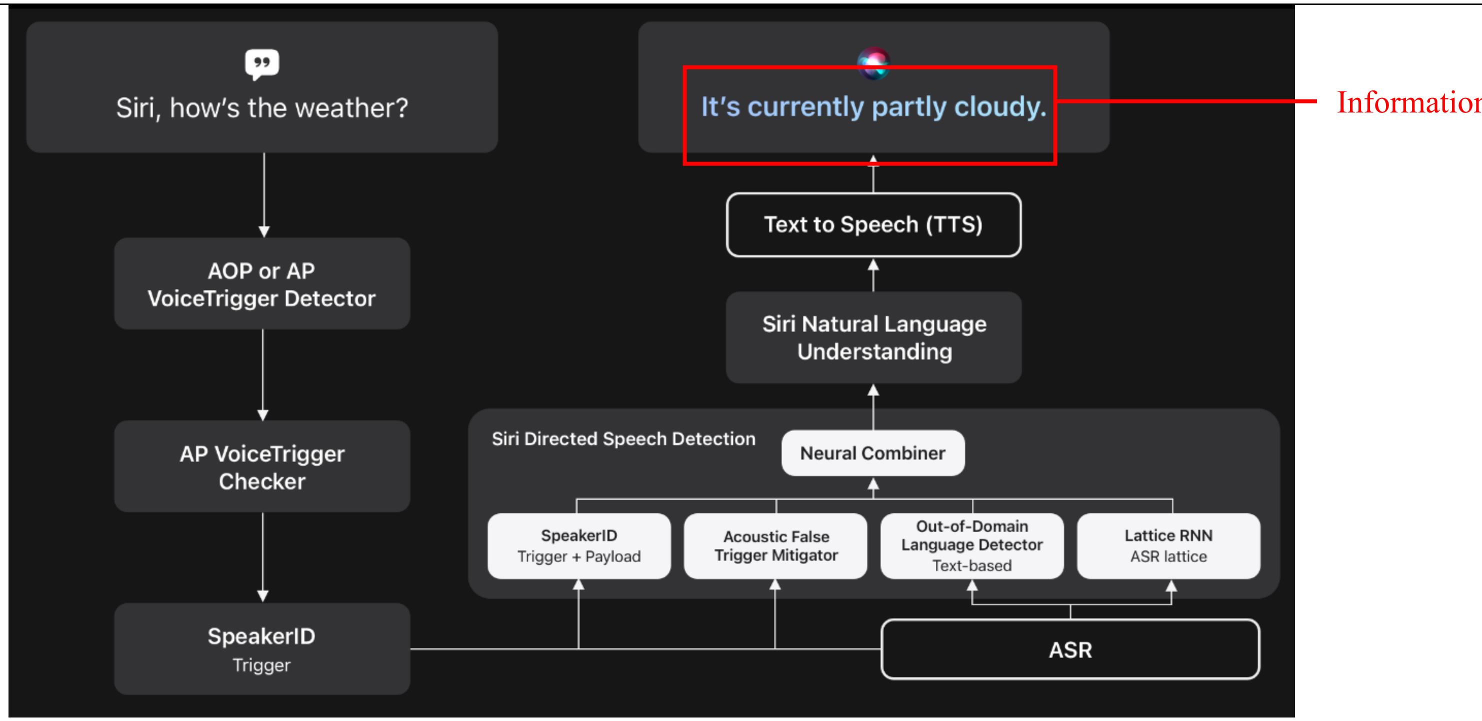
SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

discovering information

Source: https://developer.apple.com/design/human-interface-guidelines/siri (annotated)

	Being able to use Siri without pressing buttons is particularly useful when hands are busy, such as when cooking or driving, or when using the Apple Watch. As Figure 1 shows, the whole system has several parts. Most of the implementation of Siri is "in the Cloud", including the main automatic speech recognition, the natural language interpretation and the various information services. There are also servers that can provide updates to the acoustic models used by the detector. This article concentrates on the part that runs on your local device, such as an iPhone or Apple Watch. In particular, it focusses on the detector: a specialized speech recognizer which is always listening just for its wake-up phrase (on a recent iPhone with the "Hey Siri" feature enabled). Source: https://machinelearning.apple.com/research/hey-siri

	 Source: https://machinelearning.apple.coms/research/voice-trigger (annotated) Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[30.4] preparing instructions for accessing the information, the	Company provides a computer program product comprising one or more non-transitory computer readable media bearing one or more instructions for: preparing instructions for accessing the information. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, after converting audio requests to the intents, Siri provides conversational flow for system intents such that the information is retrieved. These flows help the app to fulfill the user request according to the domain of the intents.

instructions including:	Therefore, it would be apparent to a person having ordinary skill in the art that Siri prepares instructions for accessing the information. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis. Source: https://developer.apple.com/design/human-interface-guidelines/siri/ **System intents** SiriKit defines a large number of system intents that represent common tasks people do, such as playing music, sending messages to friends, and managing notes. For system intents, Siri defines the conversational flow, while your app provides the data to complete the interaction. Source: https://developer.apple.com/design/human-interface-guidelines/siri#System-intents

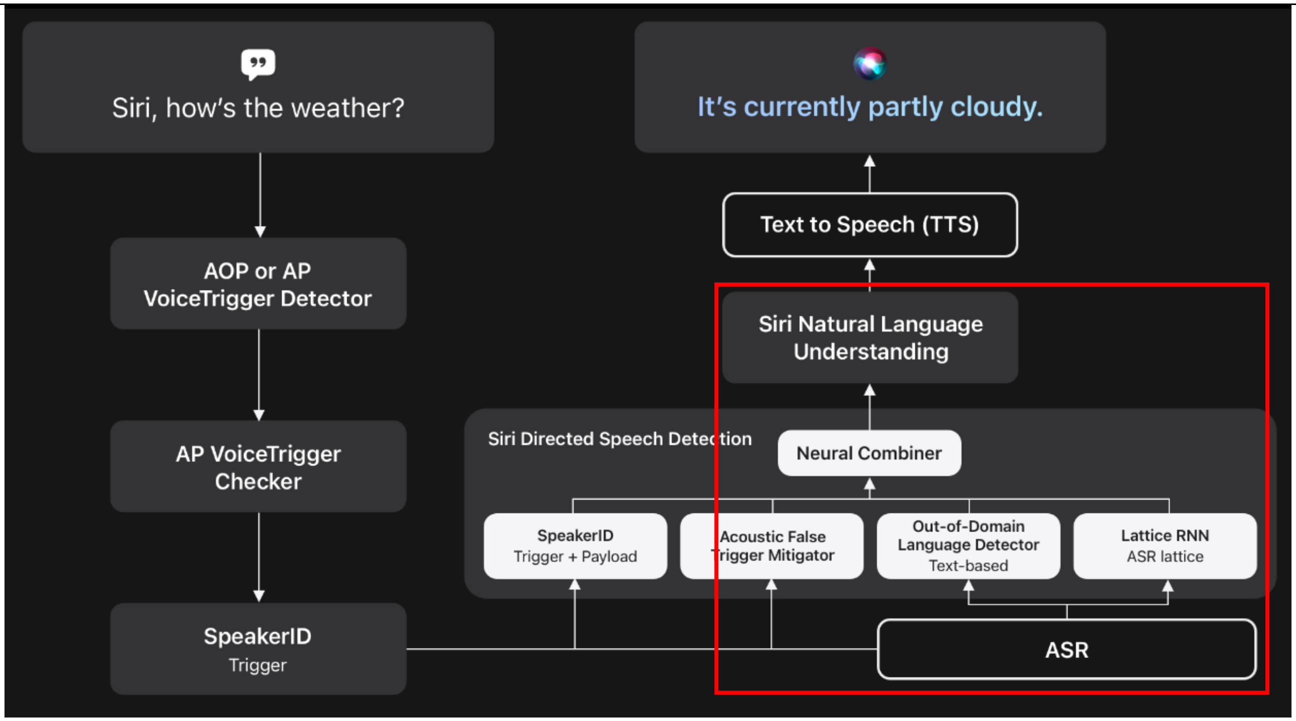
SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

Source: https://developer.apple.com/design/human-interface-guidelines/siri

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

<table>
<tr>
<td>[30.5] one or more Automatic Speech Recognition (ASR) grammar codes;</td>
<td>
Company provides a computer program product comprising one or more non-transitory computer readable media bearing one or more instructions for: preparing instructions for accessing the information, the instructions including: using one or more processing devices instructions for accessing the information, the instructions including: one or more Automatic Speech Recognition (ASR) grammar codes.
This element is infringed literally, or in the alternative, under the doctrine of equivalents.
For example, Siri does the language processing and semantic analysis to convert the requests into the intents. During semantic analysis, the audio input is matched against a grammar ("one or more Automatic Speech Recognition (ASR) grammar codes ”) to produce a semantic interpretation of the input.
A closer look at intents
When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis.
Source: https://developer.apple.com/design/human-interface-guidelines/siri/
1.4 Semantic Interpretation
A speech recognizer is capable of matching audio input against a grammar to produce a raw text transcription (also known as literal text) of the detected input. A recognizer may be capable of, but is not required to, perform subsequent processing of the raw text to produce a semantic interpretation of the input.
Source: https://www.w3.org/TR/2004/REC-speech-grammar-20040316/#S1.3
</td>
</tr>
</table>

	 Source: https://machinelearning.apple.coms/research/voice-trigger Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[30.6] one or more short text string matching codes; and	Company provides a computer program product comprising one or more non-transitory computer readable media bearing one or more instructions for: preparing instructions for accessing the information, the instructions including: one or more short text string matching codes. This element is infringed literally, or in the alternative, under the doctrine of equivalents.

For example, Siri does the natural language processing and semantic analysis to convert the requests into the intents, and provides string to match against the data. Since, the relevant information is retrieved according to the intent, upon information and belief, the instructions comprise one or more short text string matching codes.

A closer look at intents

When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis.

Source: https://developer.apple.com/design/human-interface-guidelines/siri/

Overview

Your app likely defines a number of custom types that model the data the app creates or consumes. For example, a music app might define types that represent artists, albums, and tracks. Because those types are unique to your app, the framework can't interpret them until you expose them to system services such as Siri and the Shortcuts app. *Entities* are lightweight types that provide information to the system about your app's data or concepts relating to that data. An entity identifies and queries the data it represents and describes how the system displays that data onscreen.

Source: https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents

	To let people use arbitrary text to find specific entities, adopt the `EntityStringQuery` protocol instead. Queries that adopt this protocol cause the system to display a search field above the list of suggested entities. Implement the required `entities(matching:)` function, and use the provided string to match against your data. For example, a music app might let people search for a specific album by matching against the album name. Source: https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[30.7] one or more information formatting codes operative to format a consumer device display; and	Company provides a computer program product comprising one or more non-transitory computer readable media bearing one or more instructions for: preparing instructions for accessing the information, the instructions including: one or more information formatting codes operative to format a consumer device display. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, the intents describe how the system displays ("information formatting codes operative to format a consumer device display") the data such as dates, times, and addresses. **Overview** Your app likely defines a number of custom types that model the data the app creates or consumes. For example, a music app might define types that represent artists, albums, and tracks. Because those types are unique to your app, the framework can't interpret them until you expose them to system services such as Siri and the Shortcuts app. *Entities* are lightweight types that provide information to the system about your app's data or concepts relating to that data. An entity identifies and queries the data it represents and describes how the system displays that data onscreen.

	Source: https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents Siri displays entities like dates, times, addresses and currency amounts in a nicely formatted way. This is the result of the application of a process called inverse text normalization (ITN) to the output of a core speech recognition component. To understand the important role ITN plays, consider that, without it, Siri would display "October twenty third twenty sixteen" instead of "October 23, 2016". In this work, we Source: https://machinelearning.apple.com/research/inverse-text-normal Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.
[30.8] communicating the prepared instructions.	Company provides a computer program product comprising one or more non-transitory computer readable media bearing one or more instructions for: communicating the prepared instructions. This element is infringed literally, or in the alternative, under the doctrine of equivalents. For example, Siri communicates the instructions to execute intents to the apps in the user's mobile device such as iPhone such that the relevant information is accessed. **A closer look at intents** When people use Siri to ask questions and perform actions, Siri does the language processing and semantic analysis needed to turn their requests into intents for your app to handle. The exception is the personal phrase that people create to run a shortcut: When people speak the exact phrase, Siri recognizes it without doing additional processing or analysis.

Source: https://developer.apple.com/design/human-interface-guidelines/siri/

SiriKit provides the following intents.

Domain (link to developer guidance)	Intents
VoIP Calling	Initiate calls.
Workouts	Start, pause, resume, end, and cancel workouts.
Lists and Notes	Create notes.
	Search for notes.
	Create reminders based on a date, time, or location.
Media	Search for and play media content, such as video, music, audiobooks, and podcasts.
	Like or dislike items.
	Add items to a library or playlist.

Source: https://developer.apple.com/design/human-interface-guidelines/siri

Further, to the extent this element is performed at least in part by Company's software source code, Plaintiff shall supplement these contentions pursuant to production of such source code by the Company.

2. List of References

1. https://machinelearning.apple.com/research/voice-trigger, last accessed on 8th February, 2024.
2. https://developer.apple.com/design/human-interface-guidelines/siri/, last accessed on 8th February, 2024.
3. https://machinelearning.apple.com/research/personalized-hey-siri, last accessed on 8th February, 2024.
4. https://machinelearning.apple.com/research/inverse-text-normal, last accessed on 8th February, 2024.
5. https://developer.apple.com/documentation/sirikit/inspeakable/2092309-pronunciationhint, last accessed on 8th February, 2024.
6. https://developer.apple.com/documentation/sirikit/resolution_results, last accessed on 8th February, 2024.
7. https://developer.apple.com/documentation/sirikit/media/improving_siri_media_interactions_and_app_selection, last accessed on 8th February, 2024.
8. https://arxiv.org/pdf/2005.00119.pdf, last accessed on 8th February, 2024.
9. https://support.apple.com/en-us/105020, last accessed on 8th February, 2024.
10. https://www.apple.com/siri/, last accessed on 8th February, 2024.
11. https://developer.apple.com/videos/play/tech-talks/10854/, last accessed on 8th February, 2024.
12. https://developer.apple.com/design/human-interface-guidelines/siri, last accessed on 8th February, 2024.
13. https://developer.apple.com/documentation/appintents/integrating-custom-types-into-your-intents, last accessed on 8th February, 2024.
14. https://www.w3.org/TR/2004/REC-speech-grammar-20040316/#S1.3, last accessed on 8th February, 2024.
15. https://developer.apple.com/design/human-interface-guidelines/siri#System-intents, last accessed on 8th February, 2024.